Datafly: a System for Providing Anonymity in Medical Data

Latanya Sweeney

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge,MA 02139 USA
Email: sweeney@ai.mit.edu

Abstract

We present a computer program named Datafly that maintains anonymity in medical data by automatically generalizing, substituting, inserting and removing information as appropriate without losing many of the details found within the data. Decisions are made at the field and record level at the time of database access, so the approach can be used on the fly in role-based security within an institution, and in batch mode for exporting data from an institution. Often organizations release and receive medical data with all explicit identifiers, such as name, address, phone number, and Social Security number, removed in the incorrect belief that patient confidentiality is maintained because the resulting data look anonymous; however, we show that in most of these cases, the remaining data can be used to re-identify individuals by linking or matching the data to other databases or by looking at unique characteristics found in the fields and records of the database itself. When these less apparent aspects are taken into account, each released record can be made to ambiguously map to many possible people, providing a level of anonymity which the user determines.

Keywords

Confidentiality, privacy, computational disclosure control, electronic medical records

1 INTRODUCTION

Sharing and disseminating electronic medical records while maintaining a commitment to patient confidentiality is one of the biggest challenges facing medical informatics and society at large [Tur90]. A few years ago, in 1994, we surveyed some college students at Harvard and in Taiwan. We posed the question: "Does your school have the right to read your electronic mail?" Students at Harvard (18 of 19 or 95%) stated that Harvard had no right to read their electronic mail. They argued that electronic mail was like regular mail, and since Harvard had no right to read their regular mail, Harvard had no right to read their electronic mail. Taiwanese students on the other hand, voiced an opposing opinion (16 of 17 or 94%). They felt their electronic mail reflected the school, and so the school had every right to make sure students were behaving honorably.

These findings are not surprising since they mirror the ethical systems of these two societies. It seems we map old expectations onto new technical entities, believing the new version adheres to the same social contract. In the case of electronic medical records, the public's expectations

may not be consistent with actual practice and the public may not be aware that their perceived social contract is tenuous.

In 1996, *TIME*/CNN conducted a telephone poll of 406 adults in the United States [Woo96] in which 88% replied that to the best of their knowledge, no personal medical information about themselves had ever been disclosed without their permission. In a second question, 87% said laws should be passed that prohibit health care organizations from giving out medical information without first obtaining the patient's permission.

Analysis of the detailed information contained within electronic medical records promises many advantages to society, including improvements in medical care, reduced institution costs, the development of predictive and diagnostic support systems [Coo97], and the integration of applicable data from multiple sources into a unified display for clinicians [Koh96]; but these benefits require sharing the contents of medical records with secondary viewers, such as researchers, economists, statisticians, administrators, consultants, and computer scientists, to name a few. The public would probably agree these secondary parties should know some of the information buried in the record, but such disclosure should not risk identifying patients.

Woodward makes a compelling argument that to the public, patient confidentiality implies that only people directly involved in their care will have access to their medical records and that these people will be bound by strict ethical and legal standards that prohibit further disclosure [Woo96]. The public are not likely to accept that their records are kept "confidential" if large numbers of people have access to their contents. The recent report from the National Research Council warns that as more HMOs and hospitals merge, the number of people with access increases by an order of magnitude since most of these systems allow full access to all records by any authorized person[Cla97].

As one would expect, there have been many abuses. For example, in 1995, Woodward [Woo95] cited an alarming case of a Maryland banker who cross-referenced a list of patients with cancer against a list of people who had outstanding loans at his bank and then called in the loans. Linowes and Spencer [Lin90] surveyed 87 Fortune 500 companies with a total of 3.2 million employees and found that 35% said they used medical records to make decisions about employees. The *New York Times* reported cases of snooping by insiders in large hospital computer networks[Gra97], even though the use of a simple audit trail, a list of each person who looked up a patient's record, could curtail such behavior[Cla97].

Why are identified data so available? Lincoln and Essin present major concerns over the numerous uses to which medical records are put and discuss related problems when so many demands are made for its disclosure [Lin92]. We found an evolutionary problem as well. Most electronic medical records are really two medical records in one bundle. This duality came about primarily for historical reasons. In terms of the medical record, computers were first introduced as a billing system only, and the record was basically used and controlled by administrators. Compiled for remuneration from insurance companies, these records typically included diagnosis, procedure and medication codes along with the name, address, birth date, and Social Security number for each patient. Medical billing records today usually have more than 100 such fields on each patient.

The clinical condition of the patient was maintained separately, in written form, by doctors and nurses. Some cite fear from legal retaliation and others the refusal to type on a computer keyboard as reasons for clinical information being maintained outside the computer system. In fact, even today, the "real" clinical record can often be found on index cards located in the doctor's pocket.

This trend is changing rapidly and more clinical information is routinely included in the electronic medical record, which has led to even more confusion in the social contract of patient confidentiality. In our own work, if we approach some hospitals as researchers, we must petition

970202	4973251	n
970202	7321785	y
970202	8324820	n
970203	2018492	n
970203	9353481	y
970203	3856592	n

Table 1 Possibly anonymous HIV test data.

the hospital's internal review board (IRB) to state our intentions and methodologies, then they decide whether we get data and in what form; but if we approach these same hospitals as administrative consultants, data are given to us with no IRB review. The decision is made locally and acted on.

When the clinical record joins the billing record, is the resulting electronic medical record governed by administrators, who pass it along in part to independent consultants and outside agencies as the administrators deem appropriate? Or, is it governed by the doctor-patient confidentiality contract? Who governs the records maintained by the insurance companies? Pharmaceutical companies run longitudinal studies on identified patients and providers. What happens when these records are bought and sold? What about individualized prescription records maintained by local drug stores? State governments are insisting on maintaining their own encounter-level records for cost analysis. Who should get copies and for what purposes? On the one hand, we see the possible benefits from sharing information found within the medical record and within records of secondary sources; but on the other hand, we appreciate the need for doctor-patient confidentiality. The goal of this work is to provide tools for extracting needed information from medical records while maintaining a commitment to patient confidentiality.

2 BACKGROUND

In de-identified data, all explicit identifiers, such as Social Security number, name, address and phone number, are removed, generalized or replaced with a made-up alternative; the term anonymous, however, implies that the data cannot be manipulated or linked to identify any individual. Even when information shared with secondary parties is de-identified, it is far from anonymous.

Last year, we presented the Scrub System8 which locates and replaces personally identifying information in unrestricted text. Letters between physicians and notes written by clinicians often contain nicknames, phone numbers and references to other caretakers and family members. The Scrub System found 99-100% of these references, while the straightforward approach of global search-and- replace properly located no more than 30-60% of all such references. However, the Scrub System merely de-identifies information and cannot guarantee anonymity.

There are three major difficulties in providing anonymous data. One of the problems is that

ZIP Code	Birthdate	Gender	Ethnicity
33171	7/15/71	m	Caucasian
02657	2/18/73	f	Black
20612	3/12/75	m	Asian

Table 2 De-identified data that are not anonymous

anonymity is in the eye of the beholder [Swe97]. Consider an HIV testing center located in a heavily populated community within a large metropolitan area. If Table 1 shows the results for two days, then it may not appear very anonymous if the leftmost column is the date, the middle column is the patient's phone number, and the rightmost column holds the results. An electronic phone directory can match each phone number to a name and address. Although this does not identify the specific member of the household tested, the possible choices have narrowed to a particular address.

Alternatively, if the middle column in Table 1 holds random numbers assigned to samples, then identifying individuals becomes more difficult, but we still cannot guarantee the data are anonymous. If a person with inside knowledge (e.g., a doctor, patient, nurse, attendant or even a friend of the patient) recognizes a patient and recalls the patient was the second person tested that day, then the results are not anonymous to the insider. In a similar vein, medical records distributed with a provider code assigned by an insurance company are often not anonymous, because thousands of administrators often have directories that link the provider's name, address and phone number to the assigned code.

As another example, consider Table 2. If the contents of this table are a subset of an extremely large and diverse database then the three records listed in this table may appear anonymous. Suppose the ZIP code 33171 primarily consists of a retirement community; then there are very few people of such a young age living there. Likewise, 02657 is the ZIP code for Provincetown, Massachusetts, in which we found about 5 black women living year-round. The ZIP code 20612 may have only one Asian family. In these cases, information outside the data identifies the individuals.

Most towns and cities sell locally collected census data or voter registration lists that include the date of birth, name and address of each resident. This information can be linked to medical data that include a date of birth and ZIP code, even if the names, Social Security numbers and addresses of the patients are not present. Of course, census data are usually not very accurate in college towns and areas that have large transient communities, but for much of the adult population in the United States, local census information can be used to re-identify de-identified data since other personal characteristics, such as gender, date of birth, and ZIP code, often combine uniquely to identify individuals.

The 1997 voting list for Cambridge, Massachusetts contains demographics on 54,805 voters. Of these, birth date alone can uniquely identify the name and address of 12% of the voters. We can identify 29% by just birth date and gender, 69% with only a birth date and a 5-digit ZIP code, and 97% (53,033 voters) when the full postal code and birth date are used. Clearly, the risks of re-identifying data depend both on the content of the released data and on related information available to the recipient.

SSN	Ethnicity	Birth	Sex	ZIP
819491049	Caucasian	10/23/64	m	02138
749201844	Caucasian	03/15/65	m	02139
819181496	Black	09/20/65	m	02141
859205893	Asian	10/23/65	m	02157
985820581	Black	08/24/64	m	02138

Table 3 Sample database in which Asian is a uniquely identifying characteristic

A second problem with producing anonymous data concerns unique and unusual information appearing within the data themselves [Swe97]. Consider the database shown in Table 3. It is not surprising that the Social Security number is uniquely identifying, or given the size of the database, that the birth date is also unique. To a lesser degree the ZIP code identifies individuals since it is almost unique for each record. Importantly, what may not have been known without close examination of the particulars of this database is that the designation of Asian as an ethnicity is uniquely identifying. Any single uniquely occurring value can be used to identify an individual. Remember that the unique characteristic may not be known beforehand. It could be based on diagnosis, treatment, birth year, visit date, or some other little detail or combination of details available to the memory of a patient or a doctor, or knowledge about the database from some other source.

Measuring the degree of anonymity in released data poses a third problem when producing anonymous data for practical use. The Social Security Administration (SSA) releases public-use files based on national samples with small sampling fractions (usually less than 1 in 1,000); the files contain no geographic codes, or at most regional or size-of-place designators [Ale78]. The SSA recognizes that data containing individuals with unique combinations of characteristics can be linked or matched with other data sources, so the SSA's general rule is that any subset of the data that can be defined in terms of combinations of characteristics must contain at least 5 individuals. This notion of a minimal bin size, which reflects the smallest number of individuals matching the characteristics, is quite useful in providing a degree of anonymity within data. The larger the bin size, the more anonymous the data. As the bin size increases, the number of people to whom a record may refer also increases, thereby masking the identity of the actual person.

In medical databases, the minimum bin size should be much larger than the SSA guidelines suggest. Consider these three reasons: (1) most medical databases are geographically located and so one can presume, for example, the ZIP codes of a hospital's patients; (2) the fields in a medical database provide a tremendous amount of detail and any field can be a candidate for linking to other databases in an attempt to re-identify patients; and, (3) most releases of medical data are not randomly sampled with small sampling fractions, but instead include most if not all of the database.

Determining the optimal bin size to ensure anonymity is tricky. It certainly depends on the frequencies of characteristics found within the data as well as within other sources for re-identification. In addition, the motivation and effort required to re-identify released data in cases

where virtually all possible candidates can be identified must be considered. For example, if we release data that maps each record to 10 possible people and the 10 people can be identified, then all 10 candidates may even be contacted or visited in an effort to locate the actual person. Likewise, if the mapping is 1 in 100, all 100 could be phoned since visits may then be impractical, and in a mapping of 1 in 1000, a direct mail campaign could be employed. The amount of effort the recipient is willing to spend depends on their motivation. Some medical files are quite valuable, and valuable data will merit more effort. In these cases, the minimum bin size must be further increased or the sampling fraction reduced to render these efforts useless.

Of course, the expression of anonymity most semantically consistent with our intention is simply the probability of identifying a person given the released data and other possible sources. This conditional probability depends on frequencies of characteristics (bin sizes) found within the data and the outside world. Unfortunately, this probability is very difficult to compute without omniscience. In extremely large databases like that of SSA, the database itself can be used to compute frequencies of characteristics found in the general population since it contains almost all the general population; small, specialized databases, however, must estimate these values. In the next section, we will present a computer program that generalizes data based on bin sizes and estimates. Following that, we will report results using the program and discuss its limitations.

3 METHODS

We constructed a computer program named Datafly that interfaces a user with an Oracle server which, in turn, accesses a medical database. Datafly was written using Symantec C, version 7.1 and Oracle's Pro*C Precompiler version 1.4. The Pro*C Precompiler provides mechanisms for embedding SQL commands into the C program by translating the SQL statements into calls to the runtime library SQLLIB, also by Oracle.

The Datafly System's abstract view of a database is a single table where each row corresponds to a patient record and each column holds a field of information. Even though this is a natural view for relational databases, this view also holds for non-relational databases as long as information can be retrieved by patient record and by field. Of course, a flat table as implied by this description is not practical for most medical databases. In the results section, we demonstrate Datafly working with a database schema that included multiple tables where some tables had multiple occurrences per patient; for simplicity however, we will present our findings and limit our current discussion to viewing databases as flat tables.

A user-level overview of the Datafly System begins with an original database. A user requests specific fields and records, provides a profile of the person who is to receive the data, and requests a particular anonymity floor. Datafly produces a resulting database whose information matches the anonymity level set by the user with respect to the recipient profile. The Datafly System was applied to a sample database with 5 fields: SSN, Race, Birth, Sex and ZIP. Social Security numbers were automatically replaced with made-up alternatives, birth dates were generalized to the year, and ZIP codes to the first three digits. In the next paragraphs, we discuss the information provided by the user and then we examine how Datafly produces the resulting database.

3.1 Overall bin size

The user provides Datafly with an overall anonymity level, which is a number between 0 and 1 that determines the minimum bin size allowable for each field. An anonymity level of 0 provides the original data, and a level of 1 forces Datafly to produce the most general data possible given the profile of the recipient. All other values of the anonymity level A between 0 and 1 have the following relationship to the minimum bin size required b, given the total number of records in the database N:

$$b = (r_2 - r_1) * A + r_1, \qquad \text{where} \quad r_2 > r_1$$

The variable A provides the user with a mechanism for proportionally setting the minimum anonymity level in the range r_1 to r_2. Schemes for setting r_1 and r_2 are discussed in the next subsection. The parameter b, which is based on A, is the minimum bin size for each field that the recipient will receive. Information within a field will be generalized (outliers, which are extreme values not typical of the rest of the data, may be removed) to guarantee a bin size greater than or equal to b. When we examine the resulting data, every value in each field will occur at least b times with the exception of one-to-one replacement values, as is the case with Social Security numbers. Clearly, we consider b to reflect the minimal anonymity level for the data though there are some caveats which we will discuss later. The user could specify b explicitly, but by providing A instead, the user can better understand the selected value's relation to other possible values than if the user simply entered a minimal bin size.

Consider the relationship between bin sizes and selected anonymity levels using the Cambridge voters database. As A increases, the minimum bin size increases, and in order to achieve the minimal bin size requirement, values within the birth date field, for example, are re-coded. Outliers are excluded from the released data. An anonymity level of 0.7, for example, requires at least 383 occurrences of every value in each field. To accomplish this in the birth date field, dates are re-coded to reflect only the birth year. Even after generalizing over a 12-month window, the values of 8% of the voters still do not meet the requirement so these voters are dropped from the released data.

3.2 Bin size range

In the computation of b, the parameters r_1 and r_2 specify its range. Since A can be any value between 0 and 1, inclusive, the range of b is r_1 to r_2. The Datafly System allows the user to specify r_1 and r_2 explicitly, so a consistent setting across invocations of the system allows the range of b to be independent of the size of N. We could adopt a different strategy where r_1 is a constant and r_2 increases as N increases. Examples include r_2 being the $\sqrt{N}$, also written as sqrt(N), or a factor of N, such as $(\frac{1}{100}) * N$.

Another option for r_2 available in the Datafly System is a sawtooth function where r_2 fluctuates between 100 and 999 based on the size of N. Unlike a regular sawtooth wave however, the slope of the ramp for each period decreases. Specifically, $r_2 = res * N$, such that if $10^{(k-1)} < N \leq 10^k$ and $k \neq 2$, then res is $10^{-(k-2)}$. When $k = 2$, we use $(\frac{1}{10})$. In all the Datafly System results reported herein, $r_1 = 0$ and r_2 was determined by this sawtooth function.

3.3 Linking likelihood

In addition to an overall anonymity level, the user also provides a profile of the person who receives the data by specifying for each field in the database a linking likelihood P_f, which is a value between 0 and 1 that reflects whether the recipient could have or would use information external to the database that includes data within that field for a non-trivial subset of the data; that is, for each field, the user estimates the likelihood that the recipient will use outside knowledge that includes information in that field. A linking likelihood of 0 means the information is not available outside the database, or if it is available, will not be used by the recipient. Conversely, a linking likelihood of 1 would be assigned to all fields for data being exported for public use.

For example, given the public availability of birth certificate, driver license, and census databases, birth dates, ZIP codes and gender are commonly available. The linking likelihood for these fields should be 1 for data being exported for public use, World Wide Web demonstrations and general access on the internet; however, if the recipient is the patient's caretaker within the institution, the patient has agreed to release this information to the caretaker, so the likelihood for these fields should be set to 0 to give the patient's caretaker full access to the information. Researchers bound by contractual and legal constraints that prohibit their linking of the data are trusted, so the likelihood for fields on which they could link should be set around 0.5. Datafly would then provide access to the most general, but most useful, version of the data the researcher could use. Since the linking likelihood values are set independently for each field, particular fields that are important to the recipient can have lower linking likelihood values than other requested fields in an attempt to limit generalizing the data in those fields. The overall anonymity level will still be maintained. The use of linking likelihood values therefore is most effective when fields commonly used for linking are not the same as the fields requested. Consider a database of 135,000 de-identified patients to be given to a researcher. Local census data can link with birth date, gender and ZIP code to uniquely identify patients; birth year and gender alone provide an average bin size of 3 individuals. If the recipient needs and receives only the year of birth however, the average bin size based on birth date and gender expands to 1125 people. Providing the most general information the recipient can use minimizes unnecessary risk to patient confidentiality. Using linking likelihood values for each field also supports in-house role-based security systems since recipient profiles can be pre-computed and stored for doctors, nurses, clerks, and so forth. In the next subsections, we look at how bin sizes are computed based on linking likelihood values and the overall anonymity level.

3.4 Resulting bin size per field

The linking likelihood for a particular field P_f is based on the user's profile of the recipient. It is combined with the overall anonymity level A to provide a required minimum bin size b_f for each field in cases where P_f is neither 0 nor 1:

$$b_f = b + (r_2 - r_1) * P_f + r_1 \qquad \text{where } r_2 > r_1$$

The parameter P_f specifies the likelihood that the recipient will have (and use) a database that could reduce the effect of b by possibly linking on field f. The role of P_f then is to restore the effective bin size by forcing the field to adhere to a larger value. P_f is not the percentage of records believed to be identifiable by linking, but reflects the likelihood the recipient will use the

field for linking. If P_f is 0, then the original contents of the field remain, subject to the overall anonymity level. If P_f is 1, then the most generalized data across combinations of all such fields are returned, as we will discuss shortly. The choices for r_1 and r_2 may be different from those used to determine b, but it is not necessarily the case. In all the results reported herein, b and b_f are calculated with $r_1 = 0$ and r_2 determined by the sawtooth function described previously.

3.5 Replacement algorithms

As in the Scrub System [Swe96], each entity, or in this case each field, has an algorithm that is responsible for producing replacement values. These replacement algorithms are quite diverse and offer a range of options. Each replacement or re-coding algorithm is given an iteration number and returns a Boolean value denoting whether the highest-level replacement has been performed. Dates, for example, have a range of generalizations before reverting all dates to one value. A date can be lumped to the first of the month, or the quarter, or the year (to name a few possible generalizations). Datafly would try each of these in turn in an attempt to provide the required bin size.

Repeat the following until the replacement algorithm asserts level is achieved:
1. Count the number of occurrences of each type of value in the field and store the counts in an array called BinSizes and the number of counts in Total.
2. Sort the BinSizes array.
3. Let $sum = 0$
   For $i = 1$ *to* Total **do**
      if (BinSizes[i] $< b_f$) and ($sum > loss * N$)
         then $sum = sum + count$
         else break and **go to** next step
4. **If** $sum = N$ **then** suppress entire field and exit.
   If ($sum > loss * N$)
      then generalize values in the field using field's replacement algorithm
      else assert "bin level has been achieved."

Figure 1 The Datafly Algorithm

The Datafly algorithm provides an overview of the Datafly System's operation on fields whose data can be generalized. The loss value (normally set at 10% but modifiable by the user) is the maximum percentage of the total number of records in the field that can be suppressed from the resulting data. That is, the resulting data for each field will contain no less than $(N - loss * N)$ records. Since each field can drop as many as 10% (or loss) of the records, the entire database could be lost if say, 10 fields dropped a different 10% of the records. For this reason, there is a global limit on the number of records that can be dropped from all fields. The default values is $(2 * loss)$. If this value would be exceeded, the field with the largest number of dropped records is further generalized. On each consecutive invocation of a replacement strategy, the anonymity level increases since values in that field are re-coded to build larger bin sizes.

SSN	Ethnicity	Birth	Sex	ZIP
819181496	Black	09/20/65	m	02141
195925972	Black	02/14/65	m	02141
902750852	Black	10/23/65	f	02138
985820581	Black	08/24/65	f	02138
209559459	Black	11/7/65	f	02138
679392975	Black	12/1/65	f	02138
819491049	Caucasian	10/23/64	m	02138
749201844	*Caucasian*	*03/15/65*	*f*	*02139*
985302952	Caucasian	08/13/64	m	02139
925829252	Caucasian	05/05/64	m	02139

Table 4 Combinations of fields can isolate self- identifying records, especially when these are fields that are likely to be used to re-identify data. For example, there is only one Caucasian female, even though there are many females and Caucasians.

The overall complexity of the system is governed by sorting the field contents to determine the minimum bin size, which is $O(N \log N)$. Since the Datafly Algorithm is executed on each field, the complexity is $O(fN \log N)$ where f is the number of fields, but in most of these databases $f << N$, so the overall complexity is $O(N \log N)$. In the case of unique identifying numbers, such as Social Security numbers, the replacement algorithm replaces values with made-up alternatives which are consistently hashed from the original to provide proper identification across records. Diagnosis codes have generalizations using the International Classification of Disease (ICD-9) hierarchy. Geographic replacements for states or ZIP codes generalize to use regions and population size. Continuous variables, such as dollar amounts and clinical measurements, can be treated similarly to the categorical values described; however, their replacement algorithms must contain heuristics for determining meaningful ranges in which to classify the values. Possible replacement strategies not used in Datafly include changing singletons to median values, swapping values and inserting complementary records to boost overall bin measurements. These are ways to add noise to the data [Dun91], but were not elected in this implementation of Datafly, so that each value in the data is accurate though not necessarily as specific as the original.

3.6 Combinations of fields

Table 4 provides an example wherein there is only one occurrence of a female Caucasian even though there are many females and many Caucasians. Self-identifying records, such as this, can still be overlooked. To combat this problem, subsets of fields where $P_f = 1$, with the exception of

Ethnicity	Birth	Sex	ZIP	Binsize	% of N
Caucasian	1964	m	02138	1	10.00%
Caucasian	1965	f	02139	1	20.00%
Caucasian	1964	m	02139	2	40.00%
Black	1965	m	02141	2	60.00%
Black	1965	f	02138	4	100.00%

Table 5 Fields from Table 4 that can be generalized and have $P_f = 1$ are considered one large concatenated field. In this case the required bin size is 2 with a maximum loss of 10%, so further generalization must be done even though the Birth field has been generalized to the year.

one-to-one replacement fields, are treated as one concatenated field which must meet the minimum bin size ($b_{P_f=1}$) determined by:

$$b_{P_f=1} = \max(b, e)$$

The parameter e reflects the bin size necessary to neutralize the effort or motivation the recipient may spend on re-identifying the fields where $P_f = 1$. This value can be set by the user, but its default value is r_2. When a subset of fields, where $P_f = 1$ each field, is considered a single field and the resulting values do not match the minimum bin size requirement, the field within the subset with the most number of bins is generalized. This process continues until the concatenated field meets the bin requirement (though outliers may be dropped).

For example, if Table 4 consists of all fields for which $P_f = 1$, then the fields Ethnicity, Birth, Sex and ZIP will be considered one concatenated field. The SSN field is not included since its replacement algorithm uses a one-for-one strategy. We further assume that the required bin size e is 2. In this case, the concatenation fails, even though the fields Ethnicity, Sex and ZIP would pass if evaluated separately. The Birth field has the largest number of bins, so its replacement algorithm is invoked. Birth would be generalized to the month which would fail, and then to the birth year. Table 5 shows the results at this intermediate stage. From there, ZIP has the most number of bins, so ZIP is generalized. The final result appears in Table 6.

We have now described the basic operation of the Datafly System. Here is a summary. The user provides an overall anonymity level A which is a value between 0 and 1. The user also provides a profile of the recipient by providing a linking likelihood P_f for each field that is also a value between 0 and 1. Based on these values an overall minimum bin size b is computed. The minimum bin size is then increased beyond b by a value based on P_f to determine the actual bin size for each field f. Subsets of fields where $P_f = 1$ are treated as one concatenated field whose minimum bin size is based on b and e. Replacement algorithms are invoked as needed to generalize the data to achieve the designated anonymity. Additional parameters are used in the computations, but these have default values that can be modified by the user. These parameters include the bin size range determined by r_1 and r_2 and a bin size related to the effort e the recipient may use to

Ethnicity	Birth	Sex	ZIP	Binsize	% of N
Caucasian	*1965*	*f*	*02100*	*1*	*10.00%*
Black	1965	m	02100	2	30.00%
Caucasian	1964	m	02100	3	60.00%
Black	1965	f	02100	4	100.00%

Table 6 The ZIP field from Table 5 has the largest number of bins, so it is generalized in order for the entire concatenation of fields to achieve the required bin size. The record containing the Caucasian female remains an outlier; it is not released.

re-identify fields where $P_f = 1$. After addressing two additional features of the Datafly System, we will report the results of applying the Datafly System to a medical record database.

3.7 Scramble the resulting records

Often computerized information is added sequentially in databases. As new patients enter the system, new visits begin, or new tests are run, the corresponding information is usually appended to the end of the appropriate table. This poses a danger of patient re-identification especially by insiders that extends beyond the information in the date field. The order in which records occur within a table can help identify a patient or patient information. To avoid this danger, the Datafly System randomly scrambles the order in which the records appear in each table before releasing the final database. This has no affect with which data is associated to which patients; it merely removes any indication of the order in which patients were processed within a particular time period. Now that we have presented an operational description of how Datafly works, in the next section, we will use Datafly to produce anonymous data from the pediatric medical database.

3.8 Multiple Records Per Patient

In addition to the Cambridge voter data described earlier, we also used a de-identified subset of a pediatric medical record system [Koh94]. It consisted of 300 patient records with 7617 visits and 285 fields stored in over 12 relational database tables. We were only concerned with fields that are commonly exported to government agencies, researchers and consultants. Table 7 lists some of the data fields used.

While the Datafly System conceptually views a database as one flat table where each row corresponds to one patient, this pediatric record system consisted of 12 tables. One table contained patient demographics in which there was one record per patient. Another table contained a list of all hospital visits, so information for most patients appeared in more than one row of this table. When multiple records are attributable to a patient and these records appear as multiple rows within a table, additional information is made available even though it is not explicitly stated within a field. The number of occurrences of patient information within the table may help identify the patient.

Hospital Patient Number
Patient Social Security Number
Patient Racial Background
Patient Birth Date
Patient Gender
Visit Date
Discharge Date
Principal Diagnosis Code (ICD9)
Procedure Codes (up to 14)
Medication Codes (up to 14)
Primary Physician ID Number
Type of Physician
Total Medication Charges

Table 7 Some of the data fields used in the medical database.

As an example, suppose the recipient already has available a summary of billing records for these patients and we are now releasing clinical information, but we do not want the recipient to be able to link the released clinical data to the billing records. The visit date is a critical field. However, as we have seen, the visit date and birth dates would be properly generalized, and given our knowledge of the recipient, the P_f values for the birth date and visit date fields would be set to 1, which would force all combinations of these fields to adhere to the minimal bin size. What remains present in the released data and also in the billing summary is the number of visits per patient, which may give away the identity of a patient. In these pediatric medical records, 15% of the patients had a unique number of visits; 30% of the patients had 4 or less visits. Over 50% of the number of visits per patient were unique values.

To protect anonymity in this situation, the user provides Datafly with the fields that link one table to another. Datafly then adds additional fields to its conceptual flat table; one additional field is added for each actual table. These fields record the number of records that appear in the actual table for each patient. These additional fields are then used to bundle outliers into the data. The minimal bin size for these additional fields is termed b_{tabf} and its default value is b. The user can specify an anonymity level between 0 and 1, inclusive, called P_{tabf} that determines b_{tabf} as follows:

$$b_{tabf} = (r_2 - r_1) * P_{tabf} + r_1, \qquad \text{where} \quad r_2 > r_1$$

Unlike other fields, these bin sizes are not weighted. In cases where $P_tabi = 1$, the field is included in the combinations-of-fields computation described in the previous subsection. When the field is determined as the one to further generalize in the combination, b_{tabf} is incremented and then generalizing occurs; quite differently however from the approach used in the Datafly Algorithm. With respect to these fields, a bin is the number of times a patient's information appears in the actual table, and the bin size is the number of patients having the same bin. In

the pediatric database, 21 patients made only 1 visit; using this terminology, the bin is 1, which reflects the number of visits, and the bin size is 21, which is the number of patients having 1 visit.

---

while (smallest bin size for f) $< b_{tabf}$, **do**:
1. Sort bins with size $< b_{tabf}$ by bin size and
   store the result in the array called outset.
2. **for** each outset[i]:
   combine the fewest possible adjacent bins
   to make the $binsize_i >= b_{tabf}$.
   Drop records as indicated by the merger;
   randomly select the records to drop.

Figure 2 The Datafly Multiple Records Algorithm

The Datafly Multiple Records Algorithm explains how outliers are merged into the data by dropping records from outliers until they combine with other outliers to satisfy the minimum bin requirement b_{tabf}.

3.9 Experimental results

Numerous tests were conducted using the Datafly System to access the pediatric medical record system. Datafly processed all queries to the database. In the following paragraphs we report results across a spectrum of anonymity levels for several fields.

The Social Security number, hospital patient number and physician identifying number all used one-to-one replacement algorithms, which are one-way hashing functions. Regardless of the requested anonymity level, A, or linking likelihood value, P_f, the values in these fields were replaced one time with made-up alternatives. An encryption option is available for these fields but was not used. The biggest distinction between these methods is that the result of a one-way hashing function can be consistent but is not considered reversible. With encryption, on the other hand, the mapping is also consistent, but there exists a key which when applied to the result reveals the original.

There are a total of 300 patients, 293 unique birth dates, and 2416 unique visit dates. The birth dates ranged from 1956 to 1995, while all 7617 visits occurred from 1/1/93 to 12/31/96. It is not surprising to see that only a one month generalization window was needed for the visit date field in comparison to the 6-48 month generalization windows required in the birth date field to achieve minimal bin sizes from 3 to 27.

Categorical values contained in the diagnosis, test, procedure, medication and type of physician fields should be meaningfully generalized. We could have grouped outliers together or provided some ordering of these values and then aggregated across the ordering as needed. However, neither of these methods would have provided groupings that would have been semantically useful: the resulting database would have been difficult to understand. Instead, we built a semantic hierarchy

from the original values, then generalized the values using replacement algorithms that simply moved up a level in the hierarchy.

For example, the diagnosis field consisted of International Classification of Disease codes, commonly referred to as ICD-9 or ICD-9-CM codes11. The codes are far from being evenly distributed in the hierarchy even though the generalization hierarchy has the same number of levels of generalization for all values. There are 3 divisions at the top level, and the numbers of codes within these divisions are 6723, 677 and 230. The first division is then further divided into 17 classifications of diseases and injuries, ranging from Infectious and Parasitic Diseases to Injury and Poisoning, and the number of codes covered in these subdivisions range from 9 to 139.

We used this hierarchy to generalize the values in the diagnosis field. The distributions of the re-coded diagnosis codes at different levels of generalization was maintained. If re-coding was based on sorting the values and then building groups that optimized inclusion of values, then the resulting distribution would have fewer peaks and would appear more even. Likewise, if re-coding was based on groups that evenly divided the range of all possible values, then group boundaries would merge or further subdivide information that would normally be bundled together. These strategies unnecessarily hide useful information. Identifying groups of ICD-9 codes that characterize the data even in at the higher levels of generalization remain easy since the groupings are consistent with the way in which the standard clusters the codes.

According to the patient gender field, the database consisted of 150 males, 149 females and 1 value was unspecified. Since the minimum bin size is greater than or equal to 3, any anonymity level above 0 causes the unspecified record to be dropped from the released data since it accounts for 0.33% of the total number of records and there are not enough occurrences of the unspecified value to meet the minimum bin size. Suppose instead the database had 80 male and 20 female records; the required bin size was 25; and, we were not willing to drop more than 10% of the total records. In this case, the gender field would become generalized to one value which would force the entire field to be suppressed. This may sound drastic, but it is the result of having a minimum bin size that is substantial in size when compared to the total number of records.

Much of our discussion has centered around discrete values. However, many clinical measurements and most dollar values are continuous. In these cases, we still did not generalize these values based on automatic scaling. Instead, we provided a hierarchy of ranges that conveyed meaning in the context of the value. Consider blood pressure readings. There are natural semantic ranges for generalizing a reading as normal, above normal, high, and so on. Semantic hierarchies were used when we generalized continuous values. The results maintained descriptive significance, making the approach and outcome similar to our report on the diagnosis field.

4 DISCUSSION

We have demonstrated that the Datafly System offers a practical approach to maintaining patient confidentiality by providing the most general version of the data possible to the recipient. This approach can be incorporated into in-house, administrative, and research procedures for exporting data. The end result is a database that provides minimal linking and matching of data since records will match many possible people. In concluding, we compare Datafly to a similar system developed by Statistics Netherlands; present a measurement scheme for data quality; and then finally, explore a contractual framework for releasing medical data.

4.1 The μ-Argus System

In 1996, The European Union began funding an effort that involves statistical offices and universities from the Netherlands, Italy and the United Kingdom. The main objective of this project is to develop specialized software for disclosing data such that the identity of any individual contained in the released data cannot be recognized. Statistics Netherlands has already produced, though has not yet released, a first version of a program named μ-Argus that seeks to accomplish this goal [Hun96]. The μ-Argus program is already considered the official confidentiality software of the European community even though Statistics Netherlands admittedly considers this first version a rough draft. A presentation of the concepts on which μ-Argus is based can be found in Willenborg and De Waal [Wil96].

The program μ-Argus, like the Datafly System, makes decisions based on bin sizes, generalizes values within fields as needed, and removes extreme outlier information from the released data. The μ-Argus program is written in C++ and runs under Windows on a PC. It accepts a flat, single table file in ASCII format. Statistics Netherlands envisions future versions of the program working over multiple tables and on different computing platforms. Operation of the program is as follows.

The user provides an overall bin size and specifies which fields are sensitive by assigning a value between 0 and 3 to each field. The program then identifies rare and therefore unsafe combinations by testing 2- or 3-combinations across the fields noted by the user as being identifying. Unsafe combinations are eliminated by generalizing fields within the combination (which μ-Argus terms global re-coding) and by local cell suppression. Rather than removing entire records when one or more fields contain outlier information, as is done in the Datafly System, the μ-Argus System simply suppresses or blanks out the outlier values at the cell-level; this process is called cell suppression [Kir94]. The resulting data typically contain all the rows and columns of the original data though there may be missing values in some cell locations.

Recall Table 4 presented earlier in which there were many Caucasians and many females, but only one female Caucasian in the database. We will now step through how the μ-Argus program produces results on this data; and then, we will compare the μ-Argus program to Datafly.

The first step is to check that each identifying field adheres to the minimum bin size. Then, pairwise combinations are examined for each pair that contains the "most identifying" field (in this case, SSN) and those that contain the "more identifying" fields (in this case, birth date, sex and ZIP). Finally, 3-combinations are examined that include the "most" and "more" identifying fields. Obviously, there are many possible ways to rate these identifying fields, and unfortunately different identification ratings yield different results. The ratings presented in this example produced the most secure result using the μ-Argus program though admittedly one may argue that too many specifics remain in the data for it to be released for public use. The value of each combination is basically a bin, and the bins with occurrences less than the minimum required bin size are considered unique and termed outliers. Clearly for all combinations that include the SSN, all such combinations are unique. One value of each outlier combination must be suppressed. For optimal results, the μ-Argus program suppresses values which occur in multiple outliers where precedence is given to the value occurring most often. The responsibility of when to generalize and when to suppress lies with the user. For this reason, the μ-Argus program operates in an interactive mode so the user can see the effect of generalizing and can then select to undo the step.

We will now compare the results of these two systems. Suppressing SSN values makes little

difference in this example. However, when working with multiple tables, the ability to link data across tables within the database to the same person is lost without consistent replacement of identifiers which provide such links. In fairness to μ-Argus, the current version does not work with multiple tables and as a result it does not take into account many of these issues including the number of records per patient, etc. In the Datafly System, the generalization across all fields in a subset of fields where $P_f = 1$ ensures that the combination across all the fields will adhere to the minimal bin size. The μ-Argus program however, only checks 2 or 3 combinations; there may exist unique combinations across 4 or more fields that would not be detected. For example, executing μ-Argus on the data in Table 4 with a bon size of 2 still contains a unique record for a Caucasian male born in 1964 that lives in the 02138 ZIP code, since there are 4 characteristics that combine to make this record unique, not 2. Treating a subset of identifying fields as a single field that must adhere to the minimum bin size, as done in the Datafly System, appears to provide more secure releases of data. Further, since the number of fields, especially demographic fields, in a medical database is large, this may prove to be a serious handicap when using the μ-Argus system with medical data.

In concluding our comparison of these systems, one drawback of both systems is the determination of the proper bin size. With large governmental databases, some agencies require a bin size of 5 and others 3 with virtually no geographic information [Kir94]. Geographic identification, as we discussed much earlier, is another problem when releasing medical data since the medical institution typically services patients in its geographical area. Therefore, these bin sizes are inappropriate for most medical data. Since there is no accepted measure of disclosure risk, there is no standard which can be applied to assure that the final results are adequate. Clearly, more research is needed in this area. What is customary is to measure risk against a specific compromising technique, such as linking to known databases, that we assume the recipient is using. Several researchers have proposed mathematical measures of the risk. Most of these calculations consist of computing the conditional probability of the intruder's success [Dun87]. Certainly, producing anonymous data requires criteria against which to check resulting data and to locate sensitive values. If this is based only on the database itself, the minimum bin sizes and sampling fractions may be far from optimal and may not reflect the general population. However, researchers have developed and tested several methods for estimating the percentage of unique values in the general population based on a smaller database [Ski92]. These methods are based on subsampling techniques and equivalence class structure. In the absence of these techniques, uniqueness in the population as based on demographic fields can only be determined using population registers, such as local census data, voter registration lists, city directories, and data from motor vehicle agencies, tax assessors and real estate agencies. all lists that include patients from the database. To produce an anonymous database, a producer could use a population register to identify sensitive values within the database.

4.2 Data quality measure

We have spent a great amount of time discussing anonymity and ways it can be measured in terms of bin sizes or the number of people to whom a record may reflect. Now we consider the complement measure which reports how much information was lost due to generalization and dropping outliers. This can easily be measured in terms of entropy. We consider an inverse measure to express data quality. For each field in the original database, count the number of different bins in each field. The entropy is simply the total number of bits required to account

for all bins in all fields in all records. When generalization of a field occurs, the number of bins decreases and likewise the entropy decreases. When outliers are dropped, the values of those bins are no longer included in the total count as well. So the higher the resulting entropy relative to the original, the better the data quality.

5 CONTRACTUAL ARRANGEMENTS

Clearly, one of the biggest drawbacks to both the Datafly and μ-Argus systems is the guesswork involved in profiling sensitive fields. In the Datafly System, the real goal of the P_f values, when $P_f \neq 1$, is to provide a mechanism for weighting the minimum bin size for a field so that generalizing values is limited in an attempt to provide more specific data to the recipient.

If a particular field is important to successfully link the released database to another database the recipient holds and the user does not specify $P_f = 1$ for that field, then the Datafly System can release data less secure than what would result from μ-Argus. This is a danger with the Datafly System. This risk cannot be solely placed on the producer of the data since the producer cannot always know what the recipient holds. The obvious demographic fields, physician identifiers, and billing information fields can be consistently and reliably protected. Certainly if we set Pf=1 for all fields suspected of linking, then the released data would be quite secure, even more so than what might result from μ-Argus, and the resulting data could be released for public-use files. However, for the release of more detailed data, there are too many sources of semi-public and private information such as pharmacy records, longitudinal studies, financial records, survey responses, occupational lists, and membership lists, to account a priori for all linking possibilities.

Unless we are proactive, the proliferation of medical data may become so widespread that it will be impossible to release medical data without further breaching confidentiality. For example, the existence of rather extensive registers of business establishments in the hands of government agencies, trade associations and firms like Dunn and Bradstreet has virtually ruled out the possibility of releasing database information about businesses [Kir94].

What is needed is a contractual arrangement between the recipient and the producer to make the trust explicit and share the risk. Below are some guidelines that make it clear which fields need to be protected against linking since the recipient is required to provide such a list. Using this additional knowledge and the techniques presented in the Datafly System, the producer can best protect the anonymity of patients in data more detailed than data for public-use. It is surprising that in most releases of medical data there are no contractual arrangements to limit further dissemination or use of the data. Even in cases where there is an IRB review, no contract usually results. Further, since the harm to individuals can be extreme and irreparable and can occur without the individual's knowledge, the penalties for abuses must be stringent. Significant sanctions or penalties for improper use or conduct should apply since remedy against abuse lies outside the Datafly System and resides in contracts, laws and policies.

1. There must be a legitimate and important research or administrative purpose served by the release of the data. The recipient must identify and explain which fields in the database are needed for this purpose.
2. The recipient must be strictly and legally accountable to the producer for the security of the data and must demonstrate adequate security protection.

3. The data must be de-identified. It must contain no explicit individual identifiers nor should it contain data that would be easily associated with an individual.
4. Of the fields the recipient requests, the recipient must identify which of these fields, during the specified lifetime of the data, the recipient could link to other data the recipient will have access to, whether the recipient intends to link to such data or not. The recipient must identify those fields for which the recipient will link the data.
5. The provider should have the opportunity to review any publication of information from the data to insure that no potential disclosures are published.
6. At the conclusion of the project, and no later than some specified date, the recipient must destroy all copies of the data.
7. The recipient must not give, sell, loan, show or disseminate the data to any other parties.

ACKNOWLEDGMENTS

The author gratefully acknowledges Beverly Woodward, Ph.D., for many discussions and comments. The author also thanks Professor Pierangela Samarati at the University of Milan for discussions; Isaac Kohane, M.D. Ph.D., for the use of his sample database; and Sylvia Barrett and Patrick Thompson for editorial suggestions. We also acknowledge the continued support of Henry Leitner and Harvard University DCE. We thank the Laboratory for Computer Science at MIT for its facilities. This work has been supported by a Medical Informatics Training Grant (1 T15 LM07092) from the National Library of Medicine.

REFERENCES

[Ale78] Alexander, L. and Jabine, T. (1978) Access to social security microdata files for research and statistical purposes. Social Security Bulletin. **41** 8.

[Cla97] Clayton, P., et al. (1997) *Protecting electronic health information. National Research Council.* Washington, DC: National Academy Press.

[Coo97] Cooper, G. et al. (1997) An evaluation of machine- learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine* **9**, no. 2: 107–138.

[Dun87] Duncan, G. and Lambert, D. (1987) The risk of disclosure for microdata. Proceedings of the Bureaus of the Census Third Annual Research Conference. Washington: Bureau of the Census.

[Dun91] Duncan, G. and Mukherjee, S. (1991) Microdata disclosure limitation in statistical databases: query size and random sample query control. IEEE Symposium on Research in Security and Privacy. Oakland: IEEE**2986** :278-287.

[Gra97] Grady, D. (1997) Hospital files as open book. *The New York Times*; New York, March 12, 1997:C8.

[Hun96] Hundepool, A. and Willenborg, L. (1996) μ and Tau-argus: software for statistical disclosure control. Third International Seminar on Statistical Confidentiality. Bled.

[Isr94] Israel, R. et al. (1994) The international classification of diseases. Deaprtment of Health and Human Services Publication. (PHS) 94-1260.

[Lin92] Lincoln, T. and Essin, D. (1992) The computer-based patient record: issues of organization, security and confidentiality. Database Security. Elsevier Science Publishers (IFIP) 1-19.

[Kir94] Kirkendall, N. et al. (1994) Report on statistical disclosure limitation methodology. Statistical Policy Working Paper. Washington: Office of Management and Budget, **22**.

[Koh94] Kohane, I. (1994) Getting the data in: three-year experience with a pediatric electronic medical record system. In: Ozbolt J., ed. *Proceedings, Symposium on Computer Applications in Medical Care.* Washington, DC: Hanley & Belfus, Inc. 457-461.

[Koh96] Kohane, I., et al. (1996) Sharing electronic medical records across heterogeneous and competing institutions. In: Cimino, J., ed. *Proceedings, American Medical Informatics Association.* Washington, DC: Hanley & Belfus, Inc, 608–612.

[Lin90] Linowes, D. and Spencer, R. (1990) Privacy: the workplace issue of the '90s. *The John Marshall Law Review*, **23**, 591-620.

[Ski92] Skinner, C. and Holmes, D. (1992) Modeling population uniqueness. *Proceedings of the International Seminar on Statistical Confidentiality.* International Statistical Institute, 175–199.

[Swe96] Sweeney, L. (1996) Replacing personally-identifying information in medical records, the Scrub system. In: Cimino, J., ed. Proceedings, American Medical Informatics Association. Washington, DC: Hanley & Belfus, Inc, 1996:333-337.

[Swe97] Sweeney, L. (1997) Weaving technology and policy together to maintain confidentiality. Journal of Law, Medicine and Ethics. Boston: American Association of Law, Medicine and Ethics, **25** :98-110.

[Tur90] Turn, R. (1990) Information privacy issues for the 1990s. IEEE Symposium on Research in Security and Privacy. Oakland: IEEE**2884** :394-400.

[Wil96] Willenborg, L. and De Waal, T. (1996) *Statistical disclosure control in practice.* New York: Springer-Verlag.

[Woo95] Woodward, B. (1995) The computer-based patient record and confidentiality. *The New England Journal of Medicine*; Boston: Massachusetts Medical Society, **333** 1419–1422.

[Woo96] Woodward, B. (1996) Patient privacy in a computerized world. *1997 Medical and Health Annual 1997.* Chicago: Encyclopedia Britannica, Inc. 256-259.