Chapter 1

MEASURES OF ANONYMITY

Suresh Venkatasubramanian
School of Computing, University of Utah
suresh@cs.utah.edu

1. Introduction

In this chapter, we survey the various approaches that have been proposed to measure privacy (and the loss of privacy). Since most privacy concerns (especially those related to health-care information) are raised in the context of legal concerns, it is instructive to view privacy from a legal perspective, rather than from purely technical considerations.

It is beyond the scope of this survey to review the legal interpretations of privacy. However, one essay on privacy that appears directly relevant (and has inspired at least one paper surveyed here) is the view of privacy in terms of access that others have to us and our information, presented by Ruth Gavison. In her view, a general definition of privacy must be one that is measurable, of value, and actionable. The first property needs no explanation; the second means that the entity being considered private must be valuable, and the third property argues that from a legal perspective, only those losses of privacy are interesting that can be prosecuted.

This survey, and much of the research on privacy, concerns itself with the measuring of privacy. The second property is implicit in most discussion of measures of privacy: authors propose basic data items that are valuable and must be protected (fields in a record, background knowledge about a distribution, and so on). The third aspect of privacy is of a legal nature and is not directly relevant to our discussion here.

What is privacy?

To measure privacy, we must define it. This, in essence, is the hardest part of the problem of measuring privacy, and is the reason for the
plethora of proposed measures. Once again, we turn to Gavison for some insight. In her paper, she argues that there are three inter-related kinds of privacy: secrecy, anonymity, and solitude. Secrecy concerns information that others may gather about us. Anonymity addresses how much "in the public gaze" we are, and solitude measures the degree to which others have physical access to us. From the perspective of protecting information, solitude relates to the physical protection of data, and is again beyond the purview of this article. Secrecy and anonymity are useful ways of thinking about privacy, and we will see that measures of privacy preservation can be viewed as falling mostly into one of these two categories.

If we think of privacy as secrecy (of our information), then a loss of privacy is leakage of that information. This can be measured through various means: the probability of a data item being accessed, the change in knowledge of an adversary upon seeing the data, and so on. If we think in terms of anonymity, then privacy leakage is measured in terms of the size of the blurring accompanying the release of data: the more the blurring, the more anonymous the data.

Privacy versus Utility. It would seem that the most effective way to preserve privacy of information would be to encrypt it. Users wishing to access the data could be given keys, and this would summarily solve all privacy issues. Unfortunately, this approach does not work in a data publishing scenario, which is the primary setting for much work on privacy preservation.

The key notion here is one of utility: the goal of privacy preservation measures is to secure access to confidential information while at the same time releasing aggregate information to the public. One common example used is that of the U.S. Census. The U.S. Census wishes to publish survey data from the census so that demographers and other public policy experts can analyze trends in the general population. On the other hand, they wish to avoid releasing information that could be used to infer facts about specific individuals; the case of the AOL search query release[34] indicates the dangers of releasing data without adequately anonymizing it.

It is this idea of utility that makes cryptographic approaches to privacy preservation problematic. As Dwork points out in her overview of differential privacy[16], a typical cryptographic scenario involves two communicating parties and an adversary attempting to eavesdrop. In the scenarios we consider, the adversary is the same as the recipient of the message, making security guarantees much harder to prove.
Privacy and utility are fundamentally in tension with each other. We can achieve perfect privacy by not releasing any data, but this solution has no utility. Thus, any discussion of privacy measures is incomplete without a corresponding discussion of utility measures. Traditionally, the two concepts have been measured using different yardsticks, and we are now beginning to see attempts to unify the two notions along a common axis of measurement.

A Note on Terminology. Various terms have been used in the literature to describe privacy and privacy loss. Anonymization is a popular term, often used to describe methods like $k$-anonymity and its successors. Information loss is used by some of the information-theoretic methods, and privacy leakage is another common expression describing the loss of privacy. We will use these terms interchangeably.

Data Anonymization Methods

The measures of anonymity we discuss here are usually defined with respect to a particular data anonymization method. There are three primary methods in use today, random perturbation, generalization and suppression. In what follows, we discuss these methods.

Perhaps the most natural way of anonymizing numerical data is to perturb it. Rather than reporting a value $x$ for an attribute, we report the value $\tilde{x} = x + r$, where $r$ is a random value drawn from an appropriate (usually bias-free) distribution. One must be careful with this approach however; if the value $r$ is chosen independently each time $x$ is queried, then simple averaging will eliminate its effect. Since introducing bias would affect any statistical analysis one might wish to perform on the data, a preferred method is to fix the perturbations in advance.

If the attribute $x$ has a domain other than $\mathbb{R}$, then perturbation is more complex. As long as the data lies in a continuous metric space (like $\mathbb{R}^d$ for instance), then a perturbation is well defined. If the data is categorical however, other methods, such as deleting items and inserting other, randomly chosen items, must be employed. We will see more of such methods below.

It is often useful to distinguish between two kinds of perturbation. Input perturbation is the process of perturbing the source data itself, and returning correct answers to queries on this perturbed data. Output perturbation on the other hand perturbs the answers sent to a query, rather than modifying the input itself.

The other method for anonymizing data is generalization, which is often used in conjunction with suppression. Suppose the data domain possesses a natural hierarchical structure. For example, ZIP codes can
be thought of as the leaves of a hierarchy, where 8411* is the parent of 84117, and 84* is an ancestor of 8411*, and so on. In the presence of such a hierarchy, attributes can be generalized by replacing their values with that of their (common) parent. Again returning to the ZIP code example, ZIP codes of the form 84117, 84118, 84120 might all be replaced by the generic ZIP 841*. The degree of perturbation can then be measured in terms of the height of the resulting generalization above the leaf values.

Data suppression, very simply, is the omission of data. For example, a set of database tuples might all have ZIP code fields of the form 84117 or 84118, with the exception of a few tuples that have a ZIP code field value of 90210. In this case, the outlier tuples can be suppressed in order to construct valid and compact generalization. Another way of performing data suppression is to replace a field with a generic identifier for that field. In the above example, the ZIP code field value of 90210 might be replaced by a null value ⊥ZIP.

Another method of data anonymization that was proposed by Zhang et al. [50] is to permute the data. Given a table consisting of sensitive and identifying attributes, their approach is to permute the projection of the table consisting of the sensitive attributes; the purpose of doing this is to retain the aggregate properties of the table, while destroying the link between identifying and sensitive attributes that could lead to a privacy leakage.

A Classification Of Methods

Broadly speaking, methods for measuring privacy can be divided into three distinct categories. Early work on statistical databases measured privacy in terms of the variance of key perturbed variables: the larger the variance, the better the privacy of the perturbed data. We refer to these approaches as statistical methods.

Much of the more recent work on privacy measures starts with the observation that statistical methods are unable to quantify the idea of background information that an adversary may possess. As a consequence, researchers have employed tools from information theory and Bayesian analysis to quantify more precisely notions of information transfer and loss. We will describe these methods under the general heading of probabilistic methods.

Almost in parallel with the development of probabilistic methods, some researchers have attacked the problem of privacy from a computational angle. In short, rather than relying on statistical or probabilistic estimates for the amount of information leaked, these measures start
Measures of Anonymity

from the idea of a resource-bounded adversary, and measure privacy in terms of the amount of information accessible by such an adversary. This approach is reminiscent of cryptographic approaches, but for the reasons outlined above is substantially more difficult.

An Important Omission: Secure Multiparty Computation.
One important technique for preserving data privacy is the approach from cryptography called secure multi-party computation (SMC). The simplest version of this framework is the so-called 'Millionaires Problem' [49]:

Two millionaires wish to know who is richer; however, they do not want to find out inadvertently any additional information about each others wealth. How can they carry out such a conversation?

In general, an SMC scenario is described by $N$ clients, each of whom owns some private data, and a public function $f(x_1, \ldots, x_N)$ that needs to be computed from the shared data without any of the clients revealing their private information.

Notice that in an SMC setting, the clients are trusted, and do not trust the central server to preserve their information (otherwise they could merely transmit the required data to the server). In all the privacy-preservation settings we will consider in this article, it is the server that is trusted, and queries to the server emanate from untrusted clients. We will not address SMC-based privacy methods further.

2. Statistical Measures of Anonymity

Query Restriction

Query restriction was one of the first methods for preserving anonymity in data[22, 25, 21, 40]. For a database of size $N$, and a fixed parameter $k$, all queries that returned either fewer than $k$ or more than $N-k$ records were rejected. Query restriction anticipates $k$-anonymity, in that the method for preserving anonymity is by returning a large set of records for any query. Contrast this with data suppression; rather than deleting records, the procedure deletes queries.

It was pointed out later[13, 12, 41, 10, 41] that query restriction could be subverted by requesting a specific sequence of queries, and then combining them using simple Boolean operators, in a construction referred to as a tracker. Thus, this mechanism is not very effective.

Anonymity via Variance

Here, we start with randomly perturbed data $\tilde{x} = x + r$, as described in Section 1.0. Intuitively, the larger the perturbation, the more blurred,
and thus more protected the value is. Thus, we can measure anonymity by measuring the variance of the perturbed data. The larger the variance, the better the guarantee of anonymity, and thus one proposal by Duncan et al. [15] is to lower bound the variance for estimators of sensitive attributes. An alternative approach, used by Agrawal and Srikant [3], is to fix a confidence level and measure the length of the interval of values of the estimator that yields this confidence bound; the longer the interval, the more successful the anonymization.

Under this model, utility can be measured in a variety of ways. The Duncan et al. paper measures utility by combining the perturbation scheme with a query restriction method, and measuring the fraction of queries that are permitted after perturbation. Obviously, the larger the perturbation (measured by the variance $\sigma^2$), the larger the fraction of queries that return sets of high cardinality. This presents a natural trade-off between privacy (increased by increasing $\sigma^2$) and utility (increased by increasing the fraction of permitted queries).

The paper by Agrawal and Srikant implicitly measures utility in terms of how hard it is to reconstruct the original data distribution. They use many iterations of a Bayesian update procedure to perform this reconstruction; however the reconstruction itself provides no guarantees (in terms of distance to the true data distribution).

**Anonymity via Multiplicity**

Perturbation-based privacy works by changing the values of data items. In generalization-based privacy, the idea is to “blur” the data via generalization. The hope here is that the blurred data set will continue to provide the statistical utility that the original data provided, while preventing access to individual tuples.

The measure of privacy here is a combinatorial variant of the length-of-interval measure used in [3]. A database is said to be $k$-anonymous [42] if there is no query that can extract fewer than $k$ records from it. This is achieved by aggregating tuples along a generalization hierarchy: for example, by aggregating zip codes up to the first three digits, and so on. $k$-anonymity was first defined in the context of record linkage: can tuples from multiple databases be joined together to infer private information inaccessible from the individual sources?

The $k$-anonymity requirement means such access cannot happen, since no query returns fewer than $k$ records, and so cannot be used to isolate a single tuple containing the private information. As a method for blocking record linkage, $k$-anonymity is effective, and much research has gone
into optimizing the computations, investigating the intrinsic hardness of computing it, and generalizing it to multiple dimensions.

3. Probabilistic Measures of Anonymity

Upto this point, an information leak has been defined as the revealing of specific data in a tuple. Often though, information can be leaked even if the adversary does not gain access to a specific data item. Such attacks usually rely on knowing aggregate information about the (perturbed) source database, as well as the method of perturbation used when modifying the data.

Suppose we attempt to anonymize an attribute $X$ by perturbing it with a random value chosen uniformly from the interval $[-1, 1]$. Fixing a confidence level of 100%, and using the measure of privacy from [3], we infer that the privacy achieved by this perturbation is 2 (the length of the interval $[-1, 1]$). Suppose however that a distribution on the values of $X$ is revealed: namely, $X$ takes a value in the range $[0, 1]$ with probability $1/2$, and a value in the range $[4, 5]$ with probability $1/2$. In this case, no matter what the actual value of $X$ is, an adversary can infer from the perturbed value $\tilde{X}$ which of the two intervals of length 1 the true value of $X$ really lies in, reducing the effective privacy to at most 1.

Incorporating background information changes the focus of anonymity measurements. Rather than measuring the likelihood of some data being released, we now have to measure a far more nebulous quantity: the “amount of new information learned by an adversary” relative to the background. In order to do this, we need more precise notions of information leakage than the variance of a perturbed value.

This analysis applies irrespective of whether we do anonymization based on random perturbation or generalization. We first consider measures of anonymization that are based on perturbation schemes, following this with an examination of measures based on generalization. In both settings, the measures are probabilistic: they compute functions of distributions defined on the data.

Measures Based on Random Perturbation

Using Mutual Information The paper by Agrawal and Aggarwal[2] proposes the use of mutual information to measure leaked information. We can use the entropy $H(A)$ to encode the amount of uncertainty (and therefore the degree of privacy) in a random variable $A$. $H(A|B)$, the conditional entropy of $A$ given $B$, can be interpreted as the amount of privacy “left” in $A$ after $B$ is revealed. Since entropy is usually expressed in terms of bits of information, we will use the expression $2^{H(A)}$ to rep-
resent the measure of privacy in $A$. Using this measure, the fraction of privacy leaked by an adversary who knows $B$ can be written as

$$P(A|B) = 1 - \frac{2^{H(A|B)}}{2^{H(A)}} = 1 - 2^{-I(A;B)}$$

where $I(A;B) = H(A) - H(A|B)$ is the mutual information between the random variables $A$ and $B$.

They also develop a notion of utility measured by the statistical distance between the source distribution of data and the perturbed distribution. They also demonstrate an EM-based method for reconstructing the maximum likelihood estimate of the source distribution, and show that it converges to the correct answer (they do not address the issue of rate of convergence).

**Handling Categorical Values** The above schemes rely on the source data being numerical. For data mining applications, the relevant source data is usually categorical, consisting of collections of transactions, each transaction defined as a set of items. For example, in the typical market-basket setting, a transaction consists of a set of items purchased by a customer.

Such sets are typically represented by binary characteristic vectors. The elementary datum that requires anonymity is membership: does item $i$ belong to transaction $t$? The questions requiring utility, on the other hand, are of the form, “which patterns have reasonable support and confidence”? In such a setting, the only possible perturbation is to flip an item’s membership in a transaction, but not so often as to change the answers to questions about patterns in any significant way.

There are two ways of measuring privacy in this setting. The approach taken by Evfimievski et al. [20] is to evaluate whether an anonymization scheme leaves clues for an adversary with high probability. Specifically, they define a privacy breach one in which the probability of some property of the input data is high, conditioned on the output perturbed data.

**Definition 3.1.** An itemset $A$ causes a privacy breach of level $\rho$ if for some item $a \in A$ and some $i \in 1 \ldots N$ we have $P[a \in t_i|A \subseteq t'_i] \geq \rho$.

Here, the event “$A \subseteq t_i$” is leaking information about the event “$a \in t_i$”. Note that this measure is absolute, regardless of what the prior probability of $a \in t_i$ might have been. The perturbation method is based on randomly sampling some items of the transaction $t_i$ to keep, and buffering with elements $a \notin t_i$ at random.

The second approach, taken by Rizvi and Haritsa[38], is to measure privacy in terms of the probability of correctly reconstructing the original
Measures of Anonymity

bit, given a perturbed bit. This can be calculated using Bayes’ Theorem, and is parametrized by the probability of flipping a bit (which they set to a constant $p$). Privacy is then achieved by setting $p$ to a value that minimizes the reconstruction probability; the authors show that a wide range of values for $p$ yields acceptable privacy thresholds.

Both papers then frame utility as the problem of reconstructing itemset frequencies accurately. [20] establishes a tradeoff between utility more precisely, in terms of the probabilities $p[l \rightarrow l'] = P[\#(t' \cap A) = l' | \#(t \cap A) = l]$.

For privacy, we have to ensure that (for example) if we fix an element $a \in t$, then the set of tuples $t$ that do contain $a$ are not overly represented in the modified itemset. Specifically, in terms of an average over the size of tuple sets returned, we obtain a condition on the $p[l \rightarrow l']$. In essence, the probabilities $p[l \rightarrow l']$ encode the tradeoff between utility (or ease of reconstruction) and privacy.

Measuring Transfer Of Information  Both the above papers have the same weakness that plagued the original statistics-based anonymization works: they ignore the problem of the background knowledge attack. A related, and yet subtly different problem is that ignoring the source data distribution may yield meaningless results. For example, suppose the probability of an item occurring any particular transaction is very high. Then the probability of reconstructing its value correctly is also high, but this would not ordinarily be viewed as a leak of information. A more informative approach would be to measure the level of “surprise”: namely whether the probability $P[a \in t_i]$, increases (or decreases) dramatically, conditioned on seeing the event $A \subseteq t'_i$.

Notice that this idea is the motivation for[2]; in their paper, the mutual information $I(A; B)$ measures the transfer of information between the source and anonymized data. Evfimievski et al[19], in a followup to [20], develop a slightly different notion of information transfer, motivated by the idea that mutual information is an “averaged” measure and that for privacy preservation, worst-case bounds are more relevant.

Formally, information leakage is measured by estimating the change in probability of a property from source to distorted data. For example, given a property $Q(X)$ of the data, they say that there is a privacy breach after perturbing the data by function $R(X)$ if for some $y$,

$$P[Q(X)] \leq \rho_1, P[Q(X)|R(X) = y] \geq \rho_2$$

where $\rho_1 \ll \rho_2$.

However, ensuring that this property holds is computationally intensive. The authors show that a sufficient condition for guaranteeing no
A $(\rho_1, \rho_2)$ privacy breach is to bound the difference in probability between two different $x_i$ being mapped to a particular $y$. Formally, they propose perturbation schemes such that

$$\frac{p[x_1 \rightarrow y]}{p[x_2 \rightarrow y]} \leq \gamma$$

Intuitively, this means that if we look back from $y$, there is no easy way of telling whether the source was $x_1$ or $x_2$. The formal relation to $(\rho_1, \rho_2)$-privacy is established via this intuition.

Formally, we can rewrite

$$I(X; Y) = \sum_y p(y) \text{KL}(p(X|Y = y) | p(X))$$

The function $\text{KL}(p(X|Y = y) | p(X))$ measures the transfer distance; it asks how different the induced distribution $p(X|Y = y)$ is from the source distribution $p(X)$. The more the difference is, the less the privacy breach is. The authors propose replacing the averaging in the above expression by a max, yielding a modified notion

$$I_w(X; Y) = \max_y p(y) \text{KL}(p(X|Y = y) | p(X))$$

They then show that a $(\rho_1, \rho_2)$-privacy breach yields a lower bound on the worst-case mutual information $I_w(X; Y)$, which is what we would expect.

**More general perturbation schemes** All of the above described perturbation schemes are local: perturbations are applied independently to data items. Kargupta et al. [27] showed that the lack of correlation between perturbations can be used to attack such a privacy-preserving mechanism. Their key idea is a spectral filtering method based on computing principal components of the data transformation matrix.

Their results suggest that for more effective privacy preservation, one should consider more general perturbation schemes. It is not hard to see that a natural generalization of these perturbation schemes is a Markov-chain based approach, where an item $x$ is perturbed to item $y$ based on a transition probability $p(y|x)$. FRAPP[4] is one such scheme based on this idea. The authors show that they can express the notion of a $(\rho_1, \rho_2)$-privacy breach in terms of properties of the Markov transition matrix. Moreover, they can express the utility of this scheme in terms of the condition number of the transition matrix.
Measures of Anonymity

Measures Based on Generalization

It is possible to mount a 'background knowledge' attack on \( k \)-anonymity. For example, it is possible that all the \( k \) records returned from a particular query share the same value of some attribute. Knowing that the desired tuple is one of the \( k \) tuples, we have thus extracted a value from this tuple without needing to isolate it.

The first approach to address this problem was the work on \( \ell \)-diversity [32]. Here, the authors start with the now-familiar idea that the privacy measure should capture the change in the adversary’s world-view upon seeing the data. However, they execute this idea with an approach that is absolute. They require that the distribution of sensitive values in an aggregate have high entropy (at least \( \log \ell \)). This subsumes \( k \)-anonymity, since we can think of the probability of leakage of a single tuple in \( k \)-anonymity as \( 1/k \), and so the “entropy” of the aggregate is \( \log k \). Starting with this idea, they introduce variants of \( \ell \)-diversity that are more relaxed about disclosure, or allow one to distinguish between positive and negative disclosure, or even allow for multi-attribute disclosure measurement.

Concurrently published, the work on \( p \)-sensitive-\( k \)-anonymity[43] attempts to do the same thing, but in a more limited way, by requiring at least \( p \) distinct sensitive values in each generalization block, instead of using entropy. A variant of this idea was proposed by Wong et al. [47]; in their scheme, termed \((\alpha, k)\)-anonymity, the additional constraint imposed on the generalization is that the fractional frequency of each value in a generalization is no more than \( \alpha \). Note that this approach automatically lower bounds the entropy of the generalization by \( \log(1/\alpha) \).

Machanavajjhala et al. [32] make the point that it is difficult to model the adversary’s background knowledge; they use this argument to justify the \( \ell \)-diversity measure. One way to address this problem is to assume that the adversary has access to global statistics of the sensitive attribute in question. In this case, the goal is to make the sensitive attribute “blend in”; its distribution in the generalization should mimic its distribution in the source data.

This is the approach taken by Li, Li and the author[31]. They define a measure called \( t \)-closeness that requires that the “distance” between the distribution of a sensitive attribute in the generalized and original tables is at most \( t \).

A natural distance measure to use would be the KL-distance from the generalized to the source distribution. However, for numerical attributes, the notion of closeness must incorporate the notion of a metric on the attribute. For example, suppose that a salary field in a table is gen-
eralized to have three distinct values (20000, 21000, 22000). One might reasonably argue that this generalization leaks more information than a generalization that has the three distinct values (20000, 50000, 80000).

Computing the distance between two distributions where the underlying domains inhabit a metric space can be performed using the metric known as the earth-mover distance[39], or the Monge-Kantorovich transportation distance[24]. Formally, suppose we have two distributions \( p, q \) defined over the elements \( X \) of a metric space \( (X, d) \). Then the earth-mover distance between \( p \) and \( q \) is

\[
d_E(p, q) = \inf_{P[x'|x]} \sum_{x,x'} d(x, x') P[x'|x] p(x)
\]

subject to the constraint \( \sum_{x} P[x'|x] p(x) = q(x') \).

Intuitively, this distance is defined as the value that minimizes the transportation cost of transforming one distribution to the other, where transportation cost is measured in terms of the distance in the underlying metric space. Note that since any underlying metric can be used, this approach can be used to integrate numerical and categorical attributes, by imposing any suitable metric (based on domain generalization or other methods) on the categorical attributes.

The idea of extending the notion of diversity to numerical attributes was also considered by Zhang et al. [50]. In this paper, the notion of distance for numerical attributes is extended in a different way: the goal for the k-anonymous blocks is that the “diameter” of the range of sensitive attributes is larger than a parameter \( e \). Such a generalization is said to be \((k, e)\)-anonymous. Note that this condition makes utility difficult. If we relate this to the \( \ell \)-diversity condition of having at least \( \ell \) distinct values, this represents a natural generalization of the approach. As stated however, the approach appears to require defining a total order on the domain of the attribute; this would prevent it from being used for higher dimensional attributes sets.

Another interesting feature of the Zhang et al. method is that it considers the down-stream problem of answering aggregate queries on an anonymized database, and argues that rather than performing generalization, it might be better to perform a permutation of the data. They show that this permutation-based anonymization can answer aggregate queries more accurately than generalization-based anonymization.

Anonymizing Inferences. In all of the above measures, the data being protected is an attribute of a record, or some distributional characteristic of the data. Another approach to anonymization is to protect the possible inferences that can be made from the data; this is akin
Measures of Anonymity

to the approach taken by Evfimievski et al. [19, 20] for perturbation-based privacy. Wang et al. [45] investigate this idea in the context of generalization and suppression. A privacy template is an inference on the data, coupled with a confidence bound, and the requirement is that in the anonymized data, this inference not be valid with a confidence larger than the provided bound. In their paper, they present a scheme based on data suppression (equivalent to using a unit height generalization hierarchy) to ensure that a given set of privacy templates can be preserved.

Clustering as \( k \)-anonymity. Viewing attributes as elements of metric space and defining privacy accordingly has not been studied extensively. However, from the perspective of generalization, many papers ([7, 30, 35]) have pointed out that generalization along a domain generalization hierarchy is only one way of aggregating data. In fact, if we endow the attribute space with a metric, then the process of generalization can be viewed in general as a clustering problem on this metric space, where the appropriate measure of anonymity is applied to each cluster, rather than to each generalized group.

Such an approach has the advantage of placing different kinds of attributes on an equal footing. When anonymizing categorical attributes, generalization proceeds along a generalization hierarchy, which can be interpreted as defining a tree metric. Numerical attributes are generalized along ranges, and \( t \)-closeness works with attributes in a general metric space. By lifting all such attributes to a general metric space, generalization can happen in a uniform manner, measured in terms of the diameters of the clusters.

Strictly speaking, these methods do not introduce a new notion of privacy; however, they do extend the applicability of generalization-based privacy measures like \( k \)-anonymity and its successors.

Measuring utility in generalization-based anonymity The original \( k \)-anonymity work defines the utility of a generalized table as follows. Each cell is the result of generalizing an attribute up a certain number of levels in a generalization hierarchy. In normalized form, the “height” of a generalization ranges from 0 if the original value is used, to 1 if a completely generalized value is used (in the scheme proposed, a value of 1 corresponds to value suppression, since that is the top level of all hierarchies). The precision of a generalization scheme is then \( 1 - \) the average height of a generalization (measured over all cells). The precision is 1 if there is no generalization and is 0 if all values are generalized.
Bayardo and Agrawal ([5]) define a different utility measure for k-anonymity. In their view, a tuple that inhabits a generalized equivalence class \( E \) of size \(|E| = j, j > k\) incurs a “cost” of \( j \). A tuple that is suppressed entirely incurs a cost of \( D \), where \( D \) is the size of the entire database. Thus, the cost incurred by an anonymization is given by

\[
C = \sum_{|E| \geq k} |E|^2 + \sum_{|E| < k} |D||E|
\]

This measure is known as the discernability metric. One can also compute the average size of a generalized class as a measure of utility.[32].

Another cost measure proposed by Iyengar[26] is a misclassification metric: As above, consider the equivalence class produced by an anonymization, and charge one unit of cost for each tuple in a minority class with respect to the collection of classes. Ignore all suppressed tuples. Again, averaging this over all tuples returns the total penalty.

Once again, when we introduce a metric structure on numeric attributes, utility has to be measured differently. Zhang et al. [50] propose measuring utility by ensuring that the range of sensitive values in each group is as small as possible, subject to the privacy constraints.

Utility vs Privacy

The problems of utility and anonymity ask the same kind of question: “how much information does the anonymized data distribution reveal about the source?” For an attribute to be anonymized, we wish this quantity to be small, but for a useful attribute, we want this quantity to be large!

Most of the schemes for ensuring data anonymity focus their effort on defining measures of anonymity, while using ad hoc measures of utility. A more balanced treatment of the two notions would use similar measures for utility and anonymity, or quantify the tradeoff that must exist between the two. In the next section, we will see how this can be performed in the context of computational approaches to anonymization.

However, even in the probabilistic context, some principled approaches to utility measurement have been developed. One paper that attempts this in the context of generalization-based anonymization is the work by Kifer and Gehrke[28]. In this paper, after performing a standard anonymization, they publish carefully chosen marginals of the source data. From these marginals, they then construct a consistent maximum entropy distribution, and measure utility as the KL-distance between this distribution and the source. The remainder of the paper is devoted to methods for constructing good marginals, and reconstructing the maximum entropy extension.
Switching to perturbation-based methods, the paper by Rastogi et al. [37] provides strong tradeoffs between privacy and utility. In this work, the authors define a measure of utility in terms of the discrepancy between the value of a counting query returned by an estimator, and the true value of the counting query. Privacy is measured using the framework of Evfimievski et al. [19], in terms of the conditional probability of a tuple being present in the anonymized data, relative to the prior probability of tuple being present in the source. One of the main results in their paper is an impossibility result limiting the tradeoff between these measures of privacy and utility.

4. Computational Measures Of Anonymity

We now turn to measures of anonymity that are defined computationally: privacy statements are phrased in terms of the power of an adversary, rather than the amount of background knowledge they possess. Such an approach is attractive for a variety of reasons: measuring privacy in terms of a distance between distributions does not tell us what kinds of attacks a resource-bounded adversary can mount: in this sense, privacy measures that rely on distributional distances are overly conservative. On the other hand, it is difficult to define precisely what kind of background knowledge an adversary has, and in the absence of such information, any privacy scheme based on background information attacks is susceptible to information leakage.

The first study of anonymity in the presence of computationally bounded adversaries was carried out by Dinur and Nissim[14]. In their framework, a database consists of a sequence of bits (this is without loss of generality), and a query $q$ consists of a subset of bit positions, with the output $a_q$ being the number of 1s in the subset. Such a query can be thought of as abstracting standard aggregation queries. The anonymization procedure is represented by an algorithm that returns the (possibly modified) answer $\tilde{a}_q$ to query $q$. The utility of the anonymization is measured by a parameter $\mathcal{E}$: an anonymization is said to be within $\mathcal{E}$ perturbation if $|a_q - \tilde{a}_q| \leq \mathcal{E}$, for all $q$.

An adversary is a Turing machine that can reconstruct a constant fraction of the bits in the database with high probability, using only invocations of the query algorithm. This reconstruction can be measured by the Hamming distance between the reconstructed database and the original database; the adversary succeeds if this distance is at most $\epsilon n$.

Rather than define privacy, the authors define “non-privacy”: they say a database is $t(n)$-non-private if for all $\epsilon > 0$, there is some adversary running in time $t(n)$ that can succeed with high probability.
In this model, adversaries are surprisingly strong. The authors show that even with almost-linear perturbation, an adversary permitted to run in exponential time can break privacy. Restricting the adversary to run in polynomial time helps, but only slightly; any perturbation $E = o(\sqrt{n})$ is not enough to preserve privacy, and this is tight.

Feasibility results are hard to prove in this model: as the authors point out, an adversary, with one query, can distinguish between the databases $1^n$ and $0^n$ if it has background knowledge that these are the only two choices. A perturbation of $n/2$ would be needed to hide the database contents. One way of circumventing this is to assume that the database itself is generated from some distribution, and that the adversary is required to reveal the value of a specific bit (say, the $i^{th}$ bit) after making an arbitrary number of queries, and after being given all bits of the database except the $i^{th}$ bit.

In this setting, privacy is defined as the condition that the adversary’s reconstruction probability is at most $1/2 + \delta$. In this setting, they show that a $\sqrt{T(n)}$-perturbed database is private against all adversaries that run in time $T(n)$.

Measuring Anonymity Via Information Transfer As before, in the case of probabilistic methods, we can reformulate the anonymity question in terms of information transfer; how much does the probability of a bit being 1 (or 0) change upon anonymization?

Dwork and Nissim[18] explore this idea in the context of computationally bounded adversaries. Starting with a database $d$ represented as a Boolean matrix and drawn from a distribution $D$, we can define the prior probability $p_{ij}^0 = P[d_{ij} = 1]$. Once an adversary asks $T$ queries to the anonymized database as above, and all other values of the database are provided, we can now define the posterior probability $p_{ij}^T$ of $d_{ij}$ taking the value 1. The change in belief can be quantified by the expression $\Delta = |c(p_{ij}^T) - c(p_{ij}^0)|$, where $c(x) = \log(x/(1-x))$ is a monotonically increasing function of $x$.

This is the simplified version of their formulation. In general, we can replace the event $d_{ij} = 1$ by the more general $f(d_{i1}, d_{i2}, \ldots d_{ik}) = 1$, where $f$ is some $k$-ary Boolean function. All the above definitions translate to this more general setting. We can now define $(\delta, T(n))$-privacy as the condition that for all distributions over databases, all functions $f$, and all adversaries making $T$ queries, the probability that the maximum change of belief is more than $\delta$ is negligibly small.

As with [14], the authors show a natural tradeoff between the degree of perturbation needed, and the level of privacy achieved. Specifically, the authors show that a previously proposed algorithm SuLQ[6] achieves
Measures of Anonymity

$(\delta, T(n))$ privacy with a perturbation $E = O(\sqrt{T(n)/\delta})$. They then go on to show that under such conditions, it is possible to perform efficient and accurate data mining on the anonymized database to estimate probabilities of the form $P[\beta|\alpha]$, where $\alpha, \beta$ are two attributes.

Indistinguishability Although the above measures of privacy develop precise notions of information transfer with respect to a bounded adversary, they still require some notion of a distribution on the input databases, as well as a specific protocol followed by an adversary. To abstract the ideas underlying privacy further, Dwork et al. [17] formulate a definition of privacy inspired by Dalenius[16]: A database is private if anything learnable from it can be learned in the absence of the database.

In order to do this, they distinguish between non-interactive privacy mechanisms, where the data publisher anonymizes the data and publishes it (input perturbation), and interactive mechanisms, in which the output to queries are perturbed (output perturbation). Dwork[16] shows that in a non-interactive setting, it is impossible to achieve privacy under this definition; in other words, it is always possible to design an adversary and an auxiliary information generator such that the adversary, combining the anonymized data and the auxiliary information, can effect a privacy breach far more often than an adversary lacking access to the database can.

In the interactive setting, we can think of the interaction between the database and the adversary as a transcript. The idea of indistinguishability is that if two databases are very similar, then their transcripts with respect to an adversary should also be similar. Intuitively, this means that if an individual adds their data to a database (causing a small change), the nominal loss in privacy is very small.

The main consequence of this formulation is that it is possible to design perturbation schemes that depend only on the query functions and the error terms, and are independent of the database. Informally, the amount of perturbation required depends on the sensitivity of the query functions: the more the function can change when one input is perturbed slightly, the more perturbation the database must incur. The details of these procedures are quite technical: the reader is referred to [16, 17] for more details.

Anonymity via Isolation

Another approach to anonymization is taken by [8, 9]. The underlying principle here is isolation: a record is private if it cannot be singled out from its neighbors. Formally, they define an adversary as an algorithm
that takes an anonymized database and some auxiliary information, and outputs a single point \( q \). The adversary succeeds if a small ball around \( q \) does not contain too many points of the database. In this sense, the adversary has isolated some points of the database.  

Under this definition of a privacy breach, they then develop methods for anonymizing a database. Like the papers above, they use a differential model of privacy: an anonymization is successful if the adversary, combining the anonymization with auxiliary information, can do no better at isolation than a weaker adversary with no access to the anonymized data.

One technical problem with the idea of isolation, which the authors acknowledge, is that it can be attacked in the same way that methods like \( k \)-anonymity are attacked. If the anonymization causes many points with similar characteristics to cluster together, then even though the adversary cannot isolate a single point, it can determine some special characteristics of the data from the clustering that might not have otherwise been inferred.

5. Conclusions And New Directions

The evolution of measures of privacy, irrespective of the specific method of perturbation or class of measure, has proceeded along a standard path. The earliest measures are absolute in nature, defining an intuitive notion of privacy in terms of a measure of obfuscation. Further development occurs when the notion of background information is brought in, and this culminates in the idea of a change in adversarial information before and after the anonymized data is presented.

From the perspective of theoretical rigor, computational approaches to privacy are the most attractive. They rely on few to no modelling assumptions about adversaries, and their cryptographic flavor reinforces our belief in their overall reliability as measures of privacy. Although the actual privacy preservation methods proposed in this space are fairly simple, they do work from very simple models of the underlying database, and one question that so far remains unanswered is the degree to which these methods can be made practically effective when dealing with the intricacies of actual databases.

The most extensive attention has been paid to the probabilistic approaches to privacy measurements. \( k \)-anonymity and its successors have inspired numerous works that study not only variants of the basic measures, but systems for managing privacy, extensions to higher dimensional spaces, as well as better methods for publishing data tables. The challenge in dealing with methods deriving from \( k \)-anonymity is the ver-
itable alphabet soup of approaches that have been proposed, all varying subtly in the nature of the assumptions used. The work by Wong et al. [46] illustrates the subtleties of modelling background information; their $m$-confidentiality measure attempts to model adversaries who exploit the desire of $k$-anonymizing schemes to generate a minimal anonymization. This kind of background information is very hard to formalize and argue rigorously about, even when we consider the general framework for analyzing background information proposed by Martin et al. [33].

**New Directions**

There are two recent directions in the area of privacy preservation measures that are quite interesting and merit further study. The first addresses the problem noted earlier: the imbalance in the study of utility versus privacy. The computational approaches to privacy preservation, starting with the work of Dinur and Nissim[14], provide formal trade-offs between utility and privacy, for bounded adversaries. The work of Kifer et al. [28] on injecting utility into privacy-preservation allows for a more general measure of utility as a distance between distributions, and Rastogi[37] et al. examine the tradeoff between privacy and utility rigorously in the perturbation framework.

With a few exceptions, all of the above measures of privacy are *global*: they assume a worst-case (or average-case) measure of privacy over the entire input, or prove privacy guarantees that are independent of the specific instance of a database being anonymized. It is therefore natural to consider *personalized* privacy, where the privacy guarantee need only be accurate with respect to the specific instance being considered, or can be tuned depending on auxiliary inputs.

The technique for anonymizing inferences developed in [45] can be viewed as such a scheme: the set of inferences needing protection are supplied as part of the input, and other inferences need not be protected. In the context of $k$-anonymity, Xiao and Tao[48] propose a technique that takes as input user preferences about the level of generalization they desire for their sensitive attributes, and adapts the $k$-anonymity method to satisfy these preferences. The work on worst-case background information modelling by Martin et al. [33] assumes that the specific background knowledge possessed by an adversary is an input to the privacy-preservation algorithm. Recent work by Nissim et al. [36] revisits the indistinguishability measure[17] (which is oblivious of the specific database instance) by designing an *instance-based* property of the query function that they use to anonymize a given database.
References


[31] Li, N., Li, T., and Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In IEEE International Conference on Data Engineering (this proceedings) (2007).


[37] Rastogi, V., Hong, S., and Suciu, D. The boundary between privacy and utility in data publishing. In Koch et al. [29], pp. 531–542.


REFERENCES


