GDA Score Overview

Author: Paul Francis, MPI-SWS

Version: Jan. 2019

Introduction

The General Data Anonymization (GDA) Score is a method for measuring de-identification technologies (including pseudonymization and anonymization --- any technology that aims to protect individual data in a dataset). The GDA Score measures two things:

1. **Defense**: how well the de-identification protects individual user privacy
2. **Utility**: how much analytic value remains after anonymization

Defense and utility go hand in hand: de-identification schemes that offer stronger defense tend to have less utility. Nevertheless, for a given level of protection, some schemes offer more utility than others.

It is our expectation that the GDA Score will evolve over time, and that multiple scores will be developed. The core concept behind the GDA Score, however, is that of explicitly querying the de-identified data, and basing the defense and utility measures on the responses. With this approach, any de-identification scheme can be measured in the same way independent of the underlying technology. This is the primary strength of the GDA Score: that it works with any de-identification technology and therefore allows an apples-to-apples comparison of different approaches.

While defense and utility share this notion of querying the data, the two measures are independent. How utility is measured bears no relation to how defense is measured. In particular, utility is very much dependent on what a given analytic task is, and so an analyst may care about certain utility measures and not others. We can expect the creation of a large number of utility measures over time. By contrast, defense has nothing to do with the analytic task, but rather depends only on the data and the de-identification technology.

At this point in time, the GDA Score is based on actual attacks on working implementations of de-identification using real data. In the future analytic approaches may be developed, but for now we rely on empirical demonstrations.

Defense

The GDA Score criteria for measuring defense are those of the EU Article 29 Data Protection Working Party Opinion on Anonymization: singling-out, linkability, and inference. (Other criteria may be added over time, but for now these are the only criteria.) A defense score is produced by literally executing an attack on the de-identification system, and measuring the extent to which singling-out, linkability, or inference have been compromised by comparing what is learned from the de-identified data with the raw (not de-identified) data. There are of course many different possible attacks, some that work better than others depending on the de-identification scheme and the data being attacked. The GDA Score for defense, then, is not a single score but rather a family of scores based on the specific attack and the dataset.

The GDA Score is only as good as the corresponding attacks. In particular, a given de-identification scheme may be weaker than the GDA Score implies simply because there is a stronger attack that has not been discovered. In this sense, the GDA Score gives an upper bound on the protection offered by a de-identification scheme. The protection may be weaker than the known score (because there are unknown stronger attacks). This is the primary weakness of the GDA Score.

The expectation is that, over time, more attacks will be implemented and the GDA Score will better represent the true protection offered by the de-identification scheme.

Utility

The GDA Score for utility measures the amount of data that is lost due to de-identification, and the amount of distortion present in the remaining data. We refer to these two measures as coverage and accuracy. The basic approach to making the measure is similar to that for defense. A variety of queries are executed over the de-identified data, and the answers are compared with the same queries made over the raw data.

GDA Score for Defense

For any given attack, the defense score measures:

- **Susceptibility**: how susceptible the various dataset attributes are to attack (how much of the data can be attacked),
- **Confidence Improvement and Claim Probability**: the accuracy of what is learned relative to how much of the susceptible data the attacker chooses to attack,
- **Prior Knowledge**: the amount and type of prior knowledge needed by the attacker (what the attacker knows about the protected data or related external data), and
- **Work**: the amount of "work" needed to do the attack (i.e. the number of queries).

A core aspect of the defense score is that of making a true/false claim. A claim is a single statement about the data according to one of the three Article 29 criteria. The Article 29 criteria are these:

Criteria	Article 29 definition
Singling-out	which corresponds to the possibility to isolate some or all records which identify an individual in the dataset
Linkability	which is the ability to link, at least, two records concerning the same data subject or a group of data subjects (either in the same database or in two different databases)
Inference	which is the possibility to deduce, with significant probability, the value of an attribute from the values of a set of other attributes

The corresponding claims are:

Criteria	True/False Claim
Singling-out	There is a single user with attributes A, B, C, …
Linkability	A given set of one or more users in a known dataset are also in the protected dataset
Inference	All users with attributes A, B, C, … also have attribute X

As an example of singling-out, the attacker may claim that there is a single user with attributes [gender = 'male', age = 48, zipcode = 48828, lastname = 'Ng']. This claim is false if there are zero such users or more than one such user, and true otherwise. The attributes don't need to be personal attributes as in this example. The attacker may for instance claim that there is a single person with the attributes [long = 44.4401, lat = 7.7491, time = '2016-11-28 17:14:22'].

The linkability claim requires the existence of two datasets, one that is publicly known, and one that is protected by the de-identification scheme. In computing the defense score, we assume that the public dataset and the protected dataset share some fraction of users. If a user in the public dataset is also in the protected dataset, then all data for that user is in both datasets.

In the course of an attack, the attacker makes multiple claims. The attack is more effective (and the defense weaker) if a higher fraction of the claims are true. This leads to the idea of **confidence**. If 95 out of 100 claims are true, then we say that the attack has 95% confidence (or alternately 0.95). What we really care about, however, is **confidence**

improvement. Confidence improvement is the extent to which the attacker's confidence is better than a statistical guess.

This is best explained by example. Suppose that the attacker has prior knowledge that there is exactly one user with attributes [age = 48, zipcode = 48828, lastname = 'Ng'], and that the goal of the attacker is to determine the gender of this user. Suppose that 50% of the users in the dataset are male. By simply claiming that this user is male, the attacker has a 50% chance of being right. Suppose that over 100 claims of this type, the attacker is right 50 times. The confidence is then 0.5, but this is no better than a statistical guess. Therefore the confidence improvement is zero, and we can regard the de-identification system as having very strong protection against this attack.

Note that in this example, we regard gender as the **learned** attribute (regardless of whether the attribute was learned correctly or not).

Our measure for confidence improvement CI is:

$$CI = (C-S)/(1-S)$$

where C is the measured confidence, and S is the statistical confidence, i.e. the probability of a statistical guess being correct (both expressed as a value between 0 and 1). Thus if the statistical confidence is S=0.5, and the measured confidence is C=0.5, then CI=(0.5-0.5)/(1-0.5)=0. If the attacker got say 60 of 100 such claims correct, then confidence improvement would be CI=(0.6-0.5)/(1-0.5)=0.1/0.5=0.2 (or 20%).

Confidence improvement, by itself, is not a sufficient measure of the effectiveness of an attack. To see why, consider a case where a researcher thinks of a new attack, and wants to achieve high confidence improvement so that he or she can publish a paper about it in a good conference. Let's introduce the notion of an attack **attempt**. An attempt is the set of queries that and computations that produces a single claim. Suppose that the nature of the attack is that the attacker can tell from the attempt whether the resulting claim is likely to be high confidence or low confidence. The researcher can then simply choose not to make a claim for a given attempt unless a high confidence is expected.

To put numbers to this, suppose that the attacker can get a confidence improvement of 99%, but in order to do so, he or she must make 100,000 attempts for each claim. In spite of the fact that the researcher obtained a confidence improvement of 99%, we can regard the defense as strong because the vast majority of attempts led to nothing. Put another way, if a real attacker could only make one attempt on a given user, the chances that it would lead to a privacy violation for that user are very low: the user is well protected.

For this reason, we introduce the notion of **claim probability**. This is the probability that the attacker makes a claim given an attempt. If either confidence improvement or claim probability is low, then protection is good.

The distinction between susceptibility and claim probability can be confusing. Data is susceptible to attack if an attacker is willing to make an attempt to attack the data, which would only happen the attacker thought there was a chance of success. For example, suppose that there is an attack that only works on attributes of type datetime. In this case, the attacker won't even bother to make attempts on any other attribute type. All columns that are not of type datetime are therefore not susceptible to this attack.

With this background in place, we can now state more precisely what the defense score measures:

Susceptibility:	The fraction of cells in the dataset for which the attacker can make an attempt (per column or per dataset)
Claim Probability:	The ratio of attempts to claims
Confidence Improvement:	(C-S)/(1-S), where C is confidence, the ratio of correct claims to all claims, and S is statistical confidence
Prior Knowledge:	The ratio of the number of cells needed as prior knowledge to the number of cells learned (per column or per dataset)

Work: The ratio of the number of cells retrieved by the attempted queries to the number of cells learned

Interpretation and use of the GDA Defense Score

The GDA Score is not a single score, but rather a family of scores. A given de-identification scheme can have many different attacks associated with it, with each attack producing an individual score. The attacks can be executed against different datasets, resulting in a score per attack per dataset.

The GDA Score does not allow one to say that a de-identification scheme is strong for all possible attacks and all possible datasets. In the best case, the GDA Score can only indicate that a given de-identification scheme is strong for all known attacks, and for all datasets of interest. In a less good case, the GDA Score might indicate that there are for instance only a small number of attacks for which a de-identification scheme is vulnerable, or that it is only strong for some datasets or certain columns in datasets, or that a large number of queries are needed for an attack. In these cases, one must determine the risk associated with a data release. For instance, it may be that an attackable column is not sensitive, or that the authorized analysts are known not to have the required knowledge for an attack.

Attack Software Framework

The Open GDA Score Project provides a software framework for executing attacks. The framework provides an API through which the attacker (the attacking software) collects knowledge, executes attack queries and receives results, and makes claims. The framework automatically computes the GDA Score.

Sample Datasets

The Open GDA Score Project provides a collection of real datasets against which to make measurements. Though real data, the datasets themselves are already pseudonymized. We have, however, added synthetic personally identifying information (PII) such as names and addresses to the datasets since de-identification schemes like pseudonymization generally operate on PII.

GDA Score for Utility

Utility is hard to measure generically because there are many thousands of possible analyses that one may do on a given dataset, and because different people are interested in different analytic results. A given de-identification scheme may work very well for one analysis task, and very poorly for another. Indeed, de-identification is often done with a pre-determined analysis task in mind, so as to preserve the information needed for that analysis task even while effectively destroying the information that would be needed for other analysis tasks.

Nevertheless, producing a set of generic utility measures is valuable because it can convey how general a given de-identification scheme is---whether it is use-case specific of whether it can handle many different use cases. At the same time, having *only* generic measures may be unduly pessimistic regarding a de-identification scheme that is designed for only one or a few use cases (a *targeted* de-identification scheme), and where the beneficiaries of that scheme don't care whether the de-identification scheme is general or not.

As a result of this, we define two types of utility scores:

1. **Friendly**: A utility score for the particular type of analysis the de-identification scheme was designed for.
2. **Generic**: A utility score for some general type of analysis unrelated to the de-identification scheme's purpose.

The Open GDA Score Project defines a number of generic utility scores. We envision that typically the designer of a targeted de-identification scheme will design friendly utility scores for that scheme so as to show off the scheme's benefits.

We use the following two measures (though more may be defined later):

1. **Coverage:** A measure of how much data is lost by the de-identification, and
2. **Accuracy:** Of the remaining data, how accurate it is.

For both measures, the basic approach is to query both the de-identified and raw data, and compare the results.

Coverage

Depending on the de-identification scheme, data can be essentially removed from the dataset. For instance, a pseudonymization scheme may simply remove columns that contain PII. As another example, Diffix hides values that apply to fewer than a few individuals, making them in some cases invisible.

For the generic coverage measure, we label data attributes (columns) as being one of two types:

1. **Continuous**: the attribute values describes a continuous range, like real numbers or dates.
2. **Enumerative**: the attribute values do not come from a continuous range, for instance name or level of education.

For continuous attributes, if the de-identification method aggregates values or otherwise allows range queries, then the attribute is essentially covered. Accuracy may suffer, depending on how much aggregation has taken place, but nevertheless the attribute may still be queried.

For enumerative attributes, however, either aggregation is not possible, or any possible aggregation requires domain-specific knowledge, like aggregating "hobby" into categories like sports, music, art, collecting, and so on. We define the generic coverage measure for enumerative attributes as the ratio of the number of attribute values that can be recovered after de-identification to the total number of attribute values that pertain to two or more users. For example, suppose that an attribute is "hobby", and has values like "tennis", "hiking", "button collecting", and "yodeling". Suppose that there are 10000 distinct hobbies, but 1000 of them are so rare as to be practiced by one individual each, and another 2000 are rare enough that the de-identification scheme hides those hobbies. For the coverage measure, we disregard the 1000 practiced by one individual. The coverage score is then 7000/9000 = 0.78 (7000 visible values over 9000 possible values).

The reason we disregard user-unique values (the values that pertain to a single individual) is because it is reasonable to expect a de-identification scheme to hide individual values (otherwise singling-out is trivially violated). If we don't disregard user-unique values, then de-identification schemes that hide user-unique values receive a lower score than they otherwise would, and are therefore in some sense punished for doing what they should be doing in the first place.

Accuracy

Accuracy is measured by comparing answers to queries over the de-identified data with the answers for the same queries over the raw data. Of course there are thousands of queries one might make, and a "query" can be anything from asking for a simple count to running a machine learning operation.

A key generic accuracy score is composed of a set of basic descriptive queries: those that count the number of distinct users and number of rows, and do simple statistical operations like sum, average, min, and max, accompanied by basic filtering conditions (SQL 'WHERE' clause).

An accuracy score can be made friendly by limiting itself to the kinds of queries that the corresponding use case calls for. To give an example, suppose that a shopping mall wishes to know generally where its visitors live and work, say for marketing purposes, and so wishes to buy this information from a tele-communications company that holds

mobility data (the time and cell tower from which the network was accessed). The tele-communications company of course holds a great deal of information about users, either directly or through inference. For instance, exactly where the users live, whether they make payments on time, how much they use their devices, what services they use, and so on. Since the shopping mall is only buying general location data, the company suppresses all non-location data, and then groups users by zip code and reports numbers of visitors in hourly increments, rounding to the nearest 20 labeled by the zip code where they live and work.

To measure the true utility of this de-identification scheme, it would be silly to query about visitors' payments to the tele-communications company since that is not the purpose of the de-identification and that data was suppressed altogether. Assuming that the mall doesn't need time accuracy better than hourly, or location accuracy better than zip-code, it would be pointless for the utility queries to request say a histogram with 5-minute time or 100-meter location accuracy. As such, the targeted utility measure would only request histograms at 1-hour increments and for zip code locations only.

Nevertheless it is important to understand that this de-identification scheme is not general, and the generic utility measures would show this.

Utility Software Framework

The Open GDA Score Project provides a software framework for utility scores. The framework applies queries to both raw and de-identified datasets and automatically computes coverage and accuracy.