UNIVERSITAT DE BARCELONA

FUNDAMENTALS OF DATA SCIENCE MASTER'S THESIS

GR^{2}ASP: Guided Re-identification Risk AnalySis Platform

Author:
Tom ROLANDUS HAGEDOORN

Supervisor:
Dr. Francesco BONCHI
Dr. Rohit KUMAR
Dr. Jordi VITRIA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamentals of Data Science

in the

Facultat de Matemàtiques i Informàtica

September 2, 2019

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc Fundamentals of Data Science

GR^{2}ASP:
Guided Re-identification Risk AnalySis Platform

by Tom ROLANDUS HAGEDOORN

Data privacy has been gaining considerable momentum in the recent years. The combination of numerous data breaches with the increasing interest for data sharing is pushing policy makers to impose stronger regulations to protect user data. In the E.U, the GDPR, in place since since May 2018, is forcing countless small companies to de-identify their datasets. Numerous privacy policies developed in the last two decades along with several tools are available for doing so. However, both the policies and the tools are relatively complex and require the user to have strong foundations in data privacy.

In this paper, I describe the development of GR^{2}ASP, a tool aimed at guiding users through de-identifying their dataset in an intuitive manner. To do so, the user is shielded from almost all the complexity inherent to data privacy, and interacts with simplified notions. Our tool differentiates itself from state-of-the-art similar tools by providing explainable recommendations in an intuitive interface, and having a human-in-the-loop approach towards data de-identification. We therefore think that it represents a considerable improvement over currently available tools, and we expect it to be frequently used, especially in the context of the SMOOTH project for which it has been commissioned.

Acknowledgements

I would like to thank my supervisors both at UB and at Eurecat, namely Jordi Vitria, Francesco Bonchi and Rohit Kumar. Many of the decisions along the way have been taken through brainstorming with them, and their regular feedback has helped shape this project as it is now. I would also like to thank my colleague Javier Cano for doing a considerable amount of work on the the front-end development, and the Smooth project team for collaborating with me and for trusting me with this task. Finally, I would like to thank the whole Big Data team at Eurecat and especially my desk neighbour Francesco Fabbri for making my time there really enjoyable.

Contents

Chapter 1

Introduction

In recent years, the amount of data collected about users has grown significantly, and this is likely to continue (Sagiroglu and Sinanc, 2013). Despite this unprecedented growth, several aspects of data adoption have been slow to catch up. Data sharing, although increasingly important for reasons such as for collaboration and openness, has seen only a moderate increase (Tenopir et al., 2011; Ross, Lehman, and Gross, 2012). This can be linked, to a great extend, to the lack of knowledge or guidelines for handling the risk inherent to releasing or sharing data. Cases where data releases have led to privacy breaches are abundant (Berg, 2008; Wjst, 2010; Christen, 2012). Furthermore, even without data releases, numerous privacy breaches have occurred through successful attacks, such as the notorious cases of Yahoo and Equifax (Thielman, 2016; Gressin, 2017). This has helped raising the public awareness of the risk related to the ubiquity of data which, in combination with the pressure of privacy advocates, has led to stronger data privacy regulations. In the E.U, the General Data Protection Regulation (GDPR), enforced since May 2018, marks the latest governmental attempt at protecting and regulating user data. These regulations clearly define concepts such as user consent, sensitive attributes as well as defining the hefty fines incurred for breaching these regulations (Tankard, 2016). However, they provide minimal information on how to de-identify a dataset, although they advice the data curators to do so. Naive approaches based on removing attributes that directly identify a user have proven to be insufficient at assuring privacy protection (Bélanger and Crossler, 2011). Therefore, privacy policies and tools for applying them have been developed, mainly by the research community, to establish frameworks for data de-identification. These policies and tools have, however, never been able to gain a wide audience, for they require a considerable amount of data privacy knowledge in order to use them. The current situation represents thus a discrepancy between strict regulations applying to a increasingly wide audience and a lack of means to take actions in order to comply with these regulations.

To bridge this gap, Eurecat is currently working on a project, named SMOOTH, aimed at providing an intuitive GDPR compliance platform. The aim of this thesis is to develop one of the modules of this platform, namely the one specializing in analyzing the re-identification risk of datasets. We have conveniently named our tool GR^{2}ASP, for Guided Re-identification Risk Analysis Platform, to convey the intuitiveness of it. In this paper I will develop the implementation of this tool and demonstrate that its features go well beyond the imposed requirements of the project. Hereinafter, the subject of this paper will be solely GR^{2}ASP, and not the SMOOTH project, unless specified otherwise.

The remainder of this chapter elaborates on the context and motivations that

led to the development of GR^{2}ASP, with the objectives that have shaped it, as well as comparing it with the current state-of-the-art. The second chapter provides the reader with a detailed background on the concepts of data privacy that are, directly or indirectly, used in our tool. Planning is discussed in chapter three, where the objectives will be thoroughly explained and the timeline of the project will be discussed. The fourth chapter demonstrates the capabilities of GR^{2}ASP through illustrative screenshots that allow the reader to see how the objectives are reflected in the interface. In the fifth chapter, the whole development process is detailed. The implementation of the platform in its final version is discussed, as well as changes that occurred during the development, either due to optimization or to user feedback. Finally, the last chapter summarizes what has been achieved and briefly mentions potential future improvements.

1.1 Context and motivations

Recently, as mentioned above, the landscape of data privacy has evolved drastically. Public awareness of the potential risks related to data privacy has peaked, and stricter data protection regulations are now enforced in the European Union. These regulations apply to the ever increasing range of data controllers, and are associated with hefty fines for non-compliance. Data having become ubiquitous, an increasing amount of individuals with a lack of extensive data privacy knowledge find themselves in a position of data controller, without fully understanding how to take action to comply with the relatively complex new regulations. Therefore, there is now a pressing need to develop tools that allow laymen to easily comply without needing training in the field of data privacy. Hence, with this purpose in mind, Eurecat is currently implementing such a tool in partnership with other E.U institutions.

GDRP. The General Data Protection Regulation (GDPR) is a wide-ranging legislation introduced by the European Union (EU) and enforceable since May 2018. It aims at standardizing data protection and giving users greater control over their personal information. It applies to all organizations operating inside the EU that interact with personal data in almost any way. Furthermore, the regulations are applicable both for the data controllers, who decide how the data will be processed, as for the data processors, who do the actual processing (Voigt and Bussche, 2017). The GDPR considers personal data to be any information relating to an identifiable natural person. This encompasses identifying, quasi-identifying and sensitive attributes as is explained in the Section 2.1.

SMOOTH Platform. In the EU, it is estimated that 93% of enterprises have less than 10 employees (Lukács, 2005). Most of these Micro Enterprises (MEnts) fall under the GDPR, however, they often lack the expertise to comply with these regulations (Sirur, Nurse, and Webb, 2018). The SMOOTH platform is a project proposed by Eurecat, in collaboration with eleven European institutions and organizations, which got accepted for the Horizon 2020 funding program of the EU. It aims at providing an intuitive and unified cloud platform that assists MEnts to adopt and comply with the GDPR, as well as creating awareness of the importance of being compliant. The project is made up of several tasks, stretching across departments and institutions, which interact with each other and thus require significant coordination. This paper focuses on module 4, which aims at analysing

the re-identification risk of datasets. From the previous module it receives the dataset as well as additional information, as is described in Chapter 3, and it will pass on a concise risk analysis that will be displayed in the summary at the end of the platform. However, as mentioned above, other parts of the platform are outside the scope of this paper.

1.2 Constraints

Before defining the objectives of our tool, it is important that the user should be aware of the restrictions and obligations related to the project. These mainly result from the involvement of numerous stakeholders, requiring rigorous coordination, along with the sensitivity of the subject at hand, namely data privacy.

Time and Collaboration. The deadline of this thesis is end of August, whereas the deadline for the module deliverable is several months later. This implies that the timeline for the development of certain aspects of the module by collaborators does not necessarily coincide with my timeline for this thesis. A significant part of the front-end development is done by a colleague, who also has to deal with other modules and projects, and thus adds to the coordination complexity. Furthermore, although there is considerable freedom within this module for me to decide on aspects of the platform, certain decisions have to be validated by collaborators from different institutions with whom we do not meet weekly. Finally, some technical choices have to be streamlined with the ones used in other modules, such as the rest-API and the front-end environment. Adapting to these methods that are new to me, also requires extra time.

Legal. The legal department of Eurecat is in the process of trying to get MEnts to share their data for this project. However this is not trivial, as doing so with unprotected data is legally difficult given that the GDPR applies even for the development of a tool aiming at facilitating the compliance of it. This means that the platform has not been tested on real data from the MEnts, but rather on synthetic or open data. Furthermore, strong restrictions apply also to sharing the code of GR^{2}ASP. Although the back-end is implemented by myself, the front-end is partly developed by a colleague, which makes it intricate to share the code for the whole tool. The platform being developed for the E.U adds yet another layer of complexity, and does not allow us, currently, to make the code public.

1.3 Objectives

As explained above, there is currently a lack of understanding of the data protection notions amongst the individuals to which they apply. The objective of our tool is not to higher the level of the users' understanding of the intricate notions of privacy policies, at least no more than the strict minimum. Rather, we want to lower the barriers for de-identifying data through an intuitive tool that shields the user from most of the inherent complexity. The minimal requirements and objectives of GR^{2}ASP are defined by the SMOOTH project, namely to provide an easy to use re-identification risk analysis platform. However, we want to go well beyond and make GR^{2}ASP an innovative tool that distinguishes itself from similar state-of-the-art tools. To do so, it will guide the user through the process of analysing privacy risks in his

dataset and applying transformations to overcome these risks. Our tool will be developed around a few unique key features that will make it stand out. Put briefly, it provides explainable recommendations, incorporating the human-in-the-loop approach within a parameterless framework. In similar tools, the impact of data transformations on the privacy risk are unclear and not highlighted. We overcome this by providing atomic recommendations, with their respective risk measures, which clearly show to the user the relationship between any transformation and its impact on the privacy risk. To facilitate the decision making process for the user as to which recommendation to apply, our tool illustrates the quality of the measures through intuitive visualizations. Having the human-in-the-loop is an inherent feature deriving from our approach of providing recommendations, but it is simultaneously one of our most important novelties. Similar tools have the user define the parameters initially, then de-identify the data and only finally show the risk analysis. This gives the user little control over what transformations are applied within the constraints provided by the parameters. To overcome this, GR^{2}ASP functions in an iterative manner, which means that when a recommendation is applied, new one are provided that build on top of the previously applied ones. The parameterless feature of our tool derives somewhat from the above described features and is very much in line with our goal of providing an intuitive tool for any individual that does not have extensive knowledge of data privacy concepts. The recommendations along with their risk measures and visualizations allow the user to easily select transformations that are most aligned with his purposes for the data. Through the recommendations, the user indirectly sets the parameters of the underlying privacy policies, as these represent the atomic transformations on which the policies rely.

Furthermore, GR^{2}ASP should function both as a module of the SMOOTH platform and as a standalone application. As mentioned above, within the framework of the project, the tool receives resources from the previous module. Comprised in these are the dataset, the definition of the attributes as well as the generalization hierarchies, which will be explained in the next Chapter. Therefore, when being used on its own, our tool has to overcome the fact that these resources are not provided. It will thus include an interface for uploading the data, as well as methods for automatically creating the attribute generalization hierarchies. Regarding the type of the attributes, we will rely on assumptions that are fairly realistic and only minimally restrictive, as will be explained in Chapter 5.

1.4 State-of-the-art

I omit commercial software from the scope of similar tools, as they tend to provide little information about their functioning (Jain, Gyanchandani, and Khare, 2016). There are numerous non-commercial de-identification tools, such as the UTD Anonymization Toolbox, Cornell Anonymization Toolkit, TIAMAT, SECRETA, and μArgus, to name a few (Prasser and Kohlmayer, 2015). However, they all have certain drawbacks such as lack of interface, lack of features, little focus on risk analysis, too much complexity, and scarce documentation.

Although each of these tools has its own benefits, the leading state-of-the-art de-identification tool is the one developed by Fabian Prasser and his colleagues at the Technical University of Munich (TUM), namely the ARX Data Anonymization

Tool (Prasser et al., 2014). Its main goal is to de-identify medical data, however it performs well on different kinds of data as well. Furthermore, it is open source and thus we can, and will, use it as a library for implementing GR^{2}ASP. Since its creation in 2014, it has been regularly updated with new features issue from research of the same authors (Prasser et Al. 2016; 2017b; 2017a; 2019; Prasser, Kohlmayer, and Kuhn, 2016; Bild, Kuhn, and Prasser, 2018) and has been used in several projects by different authors (Bergeat et al., 2014; Gkoulalas-Divanis and Loukides, 2015; Kondylakis et al., 2018). ARX is an extensive de-identification and risk analysis tool, providing the user with numerous options of privacy policies, risk and utility measures, as well as a long list of other related functions. Most of the data privacy concepts available in literature are implemented in this tool, which makes it one most apt options for researchers wanting great flexibility for de-identifying their data and analyzing the risks.

Chapter 2

Background on Data Privacy

The intuitive display of GR^{2}ASP, along with the relatively straightforward metrics displayed to the user, might not make it obvious that it relies on strong theoretical foundations of data privacy. Indeed, as is mentioned when explaining the parameterless feature of our tool, the user indirectly influences the parameters of the privacy policies through interactions with intuitive notions. Therefore, the complexity inherent to data privacy is only hidden from the user, but is still there in the background. Furthermore, the metrics with which the user interacts have been derived and compared to more advanced ones. Hence, this chapter details the concepts directly used in our tool, as well as some that have been considered and that played a role in shaping our tool. This provides the reader with an understanding of what has been implemented, in perspective with what the options are. For ease of lecture and comprehension, additional information regarding the purpose of certain notions within GR^{2}ASP is provided here rather than in Chapter 5, where their implementation is detailed.

2.1 Attribute Types

For data privacy, not all attributes are considered equally. Four types are defined with respect to how much information they provide to re-identify a record, as well as the potential danger linked with inferring them from a record. Therefore, it is important to assign the type of each attribute in order for the given privacy policy to hold and for the user's data to be protected.

Identifiers. These attributes allow to uniquely identify individuals. Common examples are names, national identification numbers and phone numbers. They provide the highest risk of re-identification and should be removed as first step towards de-identifying a dataset. It has often been assumed that removing identifiers suffices in protecting data privacy, however numerous data breaches, such as the AOL search breach, have proven this assumption to be wrong (Bingisser, 2008). Identifying attributes are not taken into account by privacy policies, as it is assumed that they are, or will be, suppressed.

Quasi-identifiers. Attributes that can be combined together in order to potentially identify individuals in a database are considered Quasi-Identifiers (QI). Their definition does not imply that they should necessarily allow to uniquely identify an individual, but rather that they provide additional information which can distinguish one record from another. Furthermore, for an attribute to be a QI depends on the context as well as on the possibility of an attacker knowing or inferring this information from a different source. A few common examples are Zip

codes, Date of Birth and Nationality. In terms of re-identification risk, if there is ambiguity, it is considered safer to define the attribute as being a QI. For example, the height of a person , although not considered to be known by an attacker in most contexts, could be inferred with a certain degree of accuracy from a picture containing an item of which the size is known.

Sensitive attributes. In general, an attribute to which the user does not want to be linked is considered to be a Sensitive Attribute (SA) . However, this depends on the context and is sometimes open to discussion. The GDPR defines that data revealing racial or ethnic origin, political opinions, religious beliefs or trade union membership as well as genetic, biometric and health data are considered sensitive. Although these attributes should receive the highest level of protection, as they are potentially the most harming ones, they are generally not transformed by the privacy policies (Ghinita, Kalnis, and Tao, 2010). The reasoning behind this is that they should not be contained in any identifying dataset, and thus cannot be used for a linkage attack as explained in Section 2.2. Furthermore, keeping SAs despite the potentially high risk implies that they are important and should thus stay in their original form. Nevertheless, as is explained in Section 2.3, some privacy policies offer indirectly a certain level of protection for the SAs, while others are defined specifically for this purpose. It is important to note that, given the GDPR's especially restrictive policy towards being allowed to process SAs, we expect most use cases of GR^{2}ASP to contain only QIs.

Insensitive attributes. Attributes which do not provide any information that could be used to re-identify an individual nor any sensitive information is defined as being insensitive and will therefore not be considered or affected by the privacy policies. It is not always trivial to know what information can be inferred from an attribute and for the sake of privacy risk, precaution should prevail.

2.2 Disclosures

Re-identification, or identity disclosure, is selected as being the main focus of our tool. This is partly because of the requirements of the SMOOTH project, but mainly because if an individual is identified in a dataset, then his entire record is considered to be known, which is the worst breach that he could be affected by. From the different privacy threats, we only consider the ones that assume the attacker to be in possession of the dataset as well as being in possession of an identifying dataset which he uses to try to link the records (Heeney et al., 2011). An identifying dataset is one that contains Identifiers as well as Quasi Identifiers used for linking records. It is irrelevant for our tool to know whether the attacker obtained the datasets legally or illegally, nor can it be known how many QIs they share. Therefore, it is common practice to assume the worst case scenario (Bayardo and Agrawal, 2005). The disclosures relevant to this framework are the ones defined depending on what knowledge can be gained from a successful linkage of records. For an individual, this information can be either membership of the dataset, attributes of his record or his whole record.

Membership disclosure. This type of disclosure occurs when an attacker is able to determine whether an individual is contained or not in a dataset. It is a precursor to different kind of disclosures, and can in itself already incur a privacy risk. For

example being linked to a dataset containing only cancer patients allows an attacker to infer that the individual has or has had cancer. Given the framework defined by the SMOOTH project, our tool always assumes that such a disclosure has already occurred.

Attribute disclosure. When an attacker can infer sensitive attributes it is known as a attribute disclosure. Not all datasets have sensitive attributes, but when they do, they are by definition the most sensitive ones, and should receive the greatest protection. When an attacker can infer attributes it implies that a membership disclosure has also occurred, however it does not imply re-identification as will be explained in Section 2.3. Although this type of disclosure is not the main focus of our tool, we will describe and implement two privacy policies that offer strong protection against it.

Identity disclosure (re-identification). This is the gravest kind of disclosure and also the primary one addressed by regulations, (Schwartz, 1994). It means that an attacker is able to link an individual to a specific record, which implies that a membership disclosure has occurred and that all sensitive attributes are prone to disclosure as well. This is also the primary type of disclosure targeted by our tool. Indeed, as is implied in the name itself, GR^{2}ASP is built around the notion of analysing and protecting against re-identification risk.

It is important to note that the above described disclosures represent the threats that can be achieved through a successful linkage attack. Under the GDPR, an attack is considered successful only if it offers 100% certainty. However, in the data privacy research community it is considered a threat for an attacker to be able to even come closer to any kind of disclosure (Li, Li, and Venkatasubramanian, 2007). This means that, for example, if through a data release an attacker's certainty that an individual has cancer goes from 1% to 10%, this already represents a certain level of threat. Although quantifiable, it is difficult to include this notion in regulations or even in our tool, as a data release should inherently provide information for it to be purposeful. Furthermore, there are numerous threats that can lead to disclosures, or at least to getting closer to one, and some are intrinsically difficult to protect against in our scenario. For example, Wong et al. argue that an attacker can improve his inferences by knowing which privacy policies are applied (Wong et al., 2006). Some threats have to be acknowledged and weighted against the value of releasing the data, while others have led to the definition of new privacy policies, as will be explained in Section 2.3. Hence, the reader and the user of our tool should be aware that compliance with the GDRP does not necessarily imply that the data privacy risks are completely negligible.

2.3 Privacy Policies

A naive approach to minimizing re-identification risk is to only remove the Identifiers. However, the above mentioned data breaches have clearly showed that this provides insufficient protection. One such data breach was intentionally achieved by Sweeney as a proof of concept (2000). In his paper, he observes that, using the 1990 U.S. Census summary data, 87% of the population can be uniquely identified using the attributes *ZIP code*, *gender* and *date of birth*, which are thus QIs. Given that the data does not contain sensitive information, the identifying attributes have

not been removed. This type of dataset that contains identifiers and QIs but not SAs is referred to, in the field of data privacy, as an identifying dataset and is used to achieve linkage attacks. To prove the potential privacy risk related to his observation, he tried to link this database with a medical one that was available to him. Using the three QIs, also contained in the medical dataset, Sweeney managed to successfully link at least one individual, and thus to infer sensitive information about him.

This proof of concept attack led Sweeney and his colleague Samarati to define the notion of k-anonymity, which has obtained wide recognition in the field of computer science and has been applied to numerous scenarios (Gedik and Liu, 2007;Campan and Truta, 2008). Several privacy policies have build on top it and are considered to provide better privacy protection and to a broader range of attacks (LeFevre, DeWitt, and Ramakrishnan, 2005;LeFevre, DeWitt, and Ramakrishnan, 2006;Wong et al., 2006). However, due considerably to its intuitiveness, it is still widely used despite the growing number of alternatives. This is also the principal reason that it is the first privacy policy implemented in GR^{2}ASP. Although indirectly providing a certain level of protection against attribute disclosure, k-anonymity does not give any guarantee for doing so. Therefore, when the dataset contains SAs, our tool enforces l-diversity and t-closeness, which are stricter notions that focus on attribute disclosure.

Transformations. The policies described in this paper rely on applying, either jointly or separately, two types of data transformations, namely record suppression and attribute generalization (LeFevre, DeWitt, and Ramakrishnan, 2005). The first technique consists of suppressing records, which is simple and fast but implies that the entire value of the record is lost. The second, and more complex, techniques generalizes an attribute to a higher level. To do so, a generalization hierarchy is used that defines, for every value of the attribute, the value that generalizes it, and this for an arbitrary number of levels. Figure 2.1 show a generalization hierarchy for the numerical attribute *Age*. In this example, level 0 shows the real age, level 1 defines bins of size 20, level 2 defines bins of size 40 by combining bins of the previous level, and in the last level all values are put in one bin. The highest level of a generalization hierarchy is always denote by "*" and is equivalent to removing the attribute. From this example one can see that generalization hierarchies for a numerical attribute is fairly intuitive, and creating these does not require to understand the meaning of the attribute. It is common to see attributes such *Zipcode* be generalized by replacing the last digits with "*"s, as can be seen in Figure 2.2. However, this is simply a special case of the above described binning technique. For example, if we remove one digit per generalization level, then the bins will be of size 10^{level}. Figure 2.2 shows an small generalization hierarchy for the categorical attribute *Nationality*, which I created based on the data shown in Table 2.1. The values of level 0 are omitted as they are too numerous and are assumed to be known by the reader. From this figure one can infer that to create such a generalization hierarchy for a categorical attribute, the semantic meaning of the values has to be known. Even for this minimal example, a minimum level of geographical knowledge is required. Furthermore, these hierarchies are highly arbitrary, both in the number of levels defined and in the groupings made, unless a common reference is used such as, for example, the taxonomic classification hierarchy of animals. For countries, numerous groupings can be made according to political, economical or geographical criteria for example. The creation of generalization hierarchies, for numerical and categorical attributes, will be further

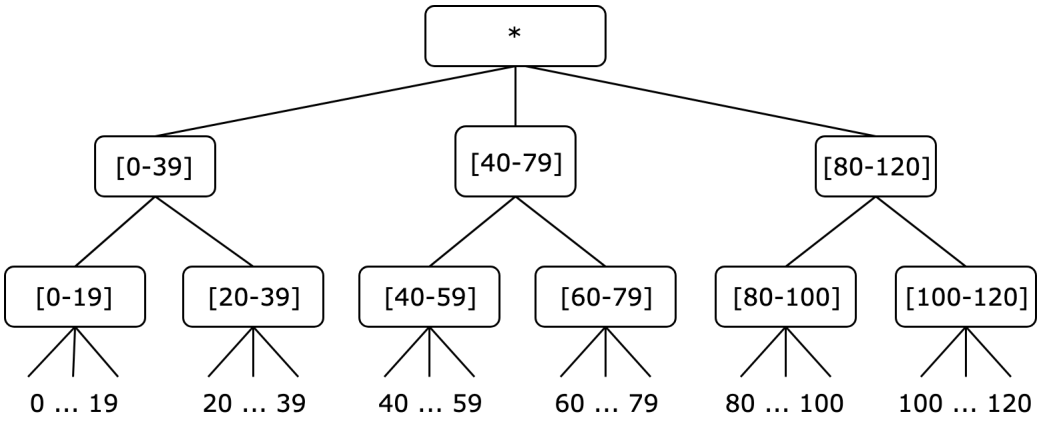

FIGURE 2.1: Generalization hierarchy for the numerical attribute *Age*

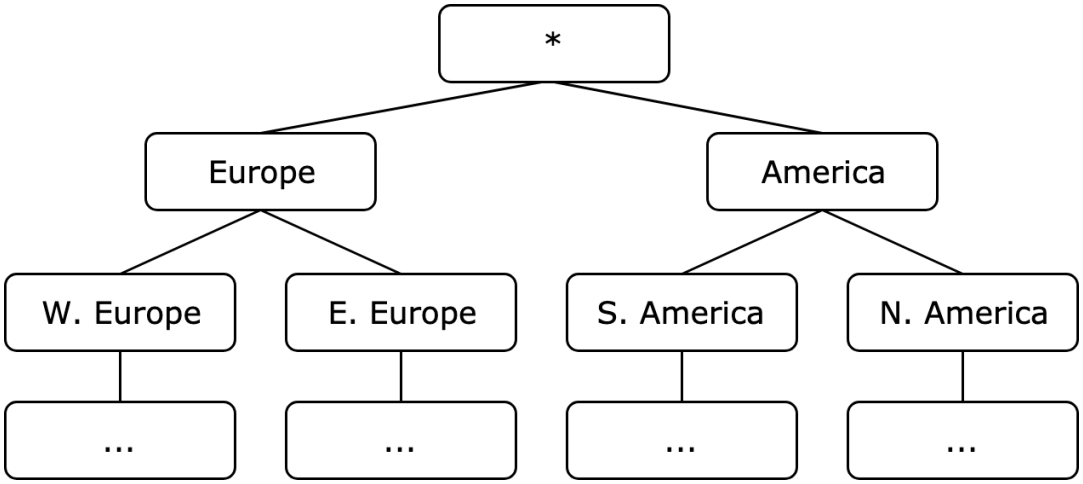

FIGURE 2.2: Generalization hierarchy for the categorical attribute *Nationality*

explained in Chapter 5, while in the meantime they are assumed to be provided to GR^{2}ASP. It is important to note that attribute generalizations can be either global or local (He and Naughton, 2009). In the first case, an attribute is generalized to the same level for all the records, whereas in the latter case, the level of generalization can be different from one record to another. However, using local generalizations is not possible within the framework of our tool and therefore, hereinafter, we only consider global generalizations.

Table 2.1 shows an example dataset of medical data that will be used to illustrate the three different privacy policies. Indeed, for each policy a transformed version of the table will be displayed and will subsequently be used to illustrate the privacy flaw that has led next policy to be stricter.

2.3.1 *k*-anonymity

The intuitive idea of hiding in a crowd is the core concept of *k*-anonymity. If an individual is the only one having a certain set of features, then it is easy to pick him out. However, if the individual is amongst a crowd all sharing the same features, he becomes indistinguishable. More formally, a database is said to be *k*-anonymous, if for

	Quasi-Identifiers			Sensitive Attributes	
	Zipcode	Age	Nationality	Salary	Disease
1	47692	63	Belgium	4K	Malaria
2	47615	41	USA	7K	Syphilis
3	47627	22	Chile	10K	AIDS
4	47693	70	Spain	5K	Cancer
5	47691	68	France	3K	Cancer
6	47629	27	Argentina	9K	AIDS
7	47610	56	Mexico	8K	Chlamydia
8	47612	42	Canada	11K	Cancer
9	47626	21	Peru	6k	AIDS

TABLE 2.1: Original table of patients' records

		Quasi-Identifiers			Sensitive Attributes	
		Zipcode	Age	Nationality	Salary	Disease
EC1	3	4762*	[20-39[	S. America	10K	AIDS
	6	4762*	[20-39[	S. America	9K	AIDS
	9	4762*	[20-39[	S. America	6k	AIDS
EC2	1	4769*	[60-79[	W. Europe	4K	Malaria
	4	4769*	[60-79[	W. Europe	5K	Cancer
	5	4769*	[60-79[	W. Europe	3K	Cancer
EC3	2	4761*	[40-59[	N. America	7K	Syphilis
	7	4761*	[40-59[	N. America	8K	Chlamydia
	8	4761*	[40-59[	N. America	11K	Cancer

TABLE 2.2: 3-anonymous version of Table 2.1

every record there exists at least $k-1$ other records having the same combination of QIs (Sweeney, 2002). Records that are identical for all their QIs are said to be in the same Equivalence Class (EC). Therefore, a dataset can also, alternatively, be defined as k-anonymous if its smallest EC is of size k. Table 2.2 displays a 3-anonymous version of Table 2.1. In practice, records of a same EC are not grouped together, nor are the indices of the records shown. This is done, in this table and in the subsequent ones, only for illustrative and comprehension purposes. One can see that, in this de-identified version of the data, an attacker in possession of an identifying dataset and with the knowledge that a specific individual is contained in the dataset, is not able to achieve, with certainty, an identity disclosure. Indeed, for every record there are at least 2 other records having identical values for *Zipcode*, *Age*, and *Nationality*. Although providing indirectly some protection against attribute disclosure, k-anonymity does not give any guarantees for doing so. The reason for this is that, it does not take into account the SAs, and a trivial example of this lack of protection can be seen in the same table. If an attacker knows the QIs of an individual in EC1, he can directly infer that he has AIDS, and thus successfully achieve an attribute disclosure.

2.3.2 l-diversity

As mentioned above, the fact that k-anonymity does not take into account the SAs leads to it providing relatively poor protection against attribute disclosure. In EC1 of Table 2.2, the *Disease* can be inferred due to its lack of diversity. From this observation, Machanavajjhala, et al. define l-diversity, a new privacy notion that builds on top of k-anonymity and also protects against this type of attacks (2006). Although they provide several different definitions for their policy, such as Distinct l-diversity, Entropy l-diversity, and Recursive (c, l)-diversity, we will focus only on the first one. The reason for this is that Distinct l-diversity is the only definition that is highly intuitive, which is an important guideline of our tool. Furthermore, both alternative definitions are still prone to the attacks that have led to defining a stricter policy, t-closeness, which we describe below.

Distinct l-diversity. The intuition behind Distinct l-diversity is opposite to k-anonymity. Indeed, the idea is to protect the SA of an individual by putting him in a group that has at least more than one level for that attribute. More formally, a table is said to be distinct l-diverse if every EC contains at least l distinct values for each sensitive attribute. (Machanavajjhala et al., 2006). Table 2.4 shows a distinct 2-diverse version of the Table 2.1. It is clear that the transformed version of the data is not prone to the previously described attack as every EC has at least 2 values for its SAs. For many scenarios, l-diversity with $l = 2$ provides sufficient protection against attribute disclosure (Zhou and Pei, 2011). However, this is not always the case and the policy still suffers from several drawbacks as are explained below.

Small l value. If the chosen value for l is smaller than the cardinality of the sensitive attribute, then not every value of the SA will be contained in every EC. If an individual can be linked to an EC that does not contain all values of the SA, then he can be targeted by a *Negative Disclosure*, which means that the attacker can rule out certain values of the SA of that individual. This can be seen in any of the ECs of Table 2.4, for example one can infer that an individual in EC3 does not have cancer. Negative disclosures are not always dangerous, but in some cases they can be.

Background knowledge. An attacker might have background knowledge about an individual from whom he is trying to disclose a SA. This information could allow him to rule out certain values of the SA until there is only one option left. For example in EC2 of Table 2.4, if the attacker knows that his target individual has not travelled recently, he can rule out Malaria with a high degree of certainty, and can thus infer that the individual has cancer.

Multi-attribute l-diversity. To successfully achieve attribute disclosure in a distinct l-diverse EC with only one SA, an attacker needs to rule out $l - 1$ values of the SA. However, this does not necessarily hold when distinct l-diversity is applied to a table with multiple SAs. Table 2.3 shows an EC that is distinct 3-diverse for both its SAs. In this example, if an attacker knows that an individual that is contained in the EC has not travelled recently, he can rule out the records having Malaria and can thus infer that the individual is Buddhist. To ensure that this cannot occur, Machanavajjhala, et al. argue that when applying distinct l-diversity to a SA, the other SAs have to be considered as being QIs. However, this leads to a great loss of information as it requires considerable attribute generalization and record suppression, hence it is

Quasi-Identifiers			Sensitive Attributes	
Zipcode	Age	Nationality	Religion	Disease
4762*	[20-39[	S. America	Christian	Malaria
4762*	[20-39[	S. America	Muslim	Malaria
4762*	[20-39[	S. America	Buddhist	Cancer
4762*	[20-39[	S. America	Buddhist	AIDS

TABLE 2.3: Equivalence Class that is Distinct 3-diverse for both the sensitive attributes *Religion* and *Disease*

		Quasi-Identifiers			Sensitive Attributes	
		Zipcode	Age	Nationality	Salary	Disease
EC1	3	476**	[0-45[	America	10K	AIDS
	6	476**	[0-45[	America	9K	AIDS
	8	476**	[0-45[	America	11K	Cancer
EC2	1	476**	[60-90[	Europe	4K	Malaria
	4	476**	[60-90[	Europe	5K	Cancer
	5	476**	[60-90[	Europe	3K	Cancer
EC3	2	476**	[0-45[	America	7K	Syphilis
	7	476**	[0-45[	America	8K	Chlamydia
	9	476**	[0-45[	America	6k	AIDS

TABLE 2.4: 2-diverse version of Table 2.1

not used in practice (Machanavajjhala et al., 2006).

2.3.3 *t*-closeness

Despite the great improvement in protection against attribute disclosure provided by *l*-diversity, Li et Al. argue that the policy has a few shortcomings which could be fixed (2007).

Firstly, they argue that the policy is unnecessary and difficult to apply in certain scenarios. In their given example, the unique SA is the positive or negative outcome of a medical test. Not considering the difference in sensitivity between the two values requires every EC of a distinct 2-diverse table to contain both outcomes. If only 1% of the table has a positive outcome, then there can be at most *number of records* $\times$ 1% ECs, which in turn leads to a considerable loss in information. For example, if there are 100 positives results in a dataset with 10,000 records, then there have to be exactly 100 ECs each of size 100 to fit the distinct 2-diversity policy, which is relatively restrictive.

Secondly, they posit that if the distribution of a SA within an EC differs considerably from the distribution of the SA in the whole table, it can allow an attacker to gain information. While using the same scenario as in the previous example, if an EC has as many positive as negative outcomes, an attacker would consider any individual in it to be 50% positive instead of the initial 1%. This means that if an

attacker only knows the distribution of *outcome* in the whole table, then he considers that any individual only has a probability 0.1 of having a positive outcome. However, if the knows the individual to be in an EC were half the outcomes are positive, he will then consider the individual to have a 0.5 probability of having a positive outcome. This example could be pushed further while still keeping distinct 2-diversity, which shows the potential for statistically inferring a sensitive attribute.

Their last and most important argument stems from the fact that l-diversity does not consider the semantic closeness of the values of a SA. This implies that although an EC is distinct l-diverse, the values of the SA might have similar meanings. For example, in Table 2.4, which is distinct 2-diverse, an attacker can infer that all individuals in EC2 have a relatively low income and that all individuals in EC3 have a STI, which represents a serious privacy breach.

To overcome these limitations, Li et Al. developed the notion of t-closeness (2007). The main idea of this privacy policy, although less intuitive than for the two previous ones, is to protect the SA of an individual by putting him in a group of individuals who's values for the SA resemble that of the population. Formulated differently, we do not want the values of the SA in an EC to deviate too much from the distribution, nor to be too similar to each other. More formally, the definition of t-closeness states that in every EC the distance between the distribution of a SA should differ from its distribution in the whole dataset by no more than a threshold t. Below, I define the metric that they use to calculate this threshold.

Earth Mover Distance. In order to measure the distance between two distributions, they use the Earth Mover Distance (EMD). The intuitive idea, from which this measure got its name, is that one distribution is seen as a pile of earth while the other distribution is seen as holes in the same space. The EMD measures the least amount of work needed to fill the holes, where moving a unit of earth by a unit of ground distance corresponds to a unit of work. For a rigorous explanation of the EMD, the reader is encouraged to read the in-depth overview of Rubner et Al. (2000). In the context of t-closeness, a unit of earth corresponds to the probability of a value of the SA, whereas the definition of a unit of ground distance depends on whether the SA is numerical or categorical.

For numerical attributes, the ground distance is computed using an ordered list of the values. Using this measure, the distance between two values v_i and v_j depends on the number of values between them in an list, where m represent the total number of values.

$$Ordered\ distance(v_i, v_j) = \frac{|i-j|}{m-1}$$

For categorical attributes, if the generalization hierarchy of a SA is not available, the equal distance measure is used. This metric states that the distance between any two values of the SA is defined to be 1. However, it does not to take into account the semantic closeness of values. Hence, when the generalization hierarchy is available, a more complex metric is used, namely the Hierarchical distance. Using this measure, the distance between two values v_i and v_j depends on the height of the lowest level to which both values can be generalized and the total height of the hierarchy.

$$Hierarchical\ distance(v_i, v_j) = \frac{lowest\ ancestor(v_i, v_j)}{height\ of\ hierarchy}$$

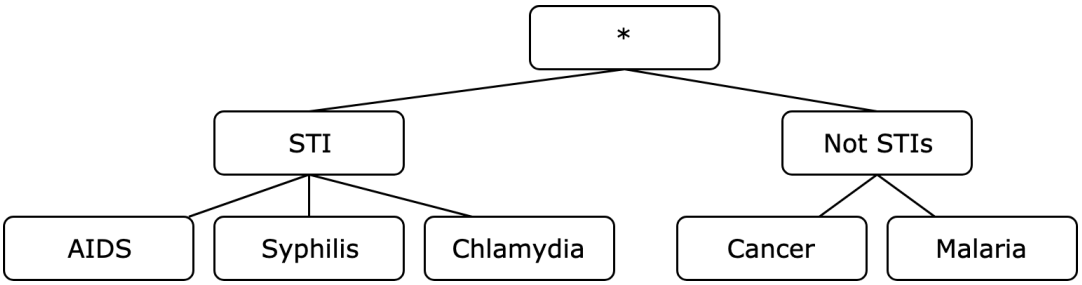

FIGURE 2.3: Generalization hierarchy for the sensitive attribute *Disease*

Where $lowest\ ancestor(v_i, v_j)$ is the lowest level of generalization to which both values can be generalized. For example, using the generalization hierarchy shown in Figure 2.3, we can compute the distance between *AIDS* and *Cancer* as being

$$distance(AIDS, Cancer) = \frac{2}{2} = 1$$

and the distance between *Cancer* and *Malaria* as being

$$distance(Cancer, Malaria) = \frac{1}{2} = 0.5$$

These examples show that when using the Hierarchical distance as the ground measure for the EDM, it highly depends on the definition of the generalization hierarchy. And, as mentioned above, the hierarchies are arbitrary and can vary greatly.

Figure 2.5 shows a version of Table 2.1 where 0.5-closeness has been applied to both SAs. We can see in this case that a similarity attack, as has been described above, cannot be achieved. None of the ECs contains only low or only high salaries, nor does any EC contain only STIs or only non-STIs. The protection provided by t-closeness, against attribute disclosure, is greater than the protection provided by distinct l-diversity. However, this comes at a significant utility cost due to requiring considerably more generalizations for the QIs, as is illustrated in this table. Given that generalization hierarchies are highly arbitrary and rarely provided, and given the high utility cost of t-closeness, we decided not to use it in GR^{2}ASP for categorical SAs. However, for numerical SAs, we think that this privacy policy makes more sense and we have thus decided to implement it only for this scenario, as will be explained in Chapter 5.

2.4 Re-identification Risk Measures

As explained in Chapter 1, the focus of GR^{2}ASP lies in analyzing the risk related to re-identification, rather than the risk related to membership or attribute disclosure. Numerous measures have been proposed and are used for doing so, however none has been selected as being the best overall. In line with our objective of simplicity, we decide to use measures that are intuitive to the user, instead of complex ones. Therefore, to quantify the re-identification risk, our tool will display visualizations of the Prosecutor Risk and the Average Equivalence Class Size risk. However, we also briefly explain other relevant metrics for the reader to better understand the

		Quasi-Identifiers			Sensitive Attributes	
		Zipcode	Age	Nationality	Salary	Disease
EC1	5	476**	*	*	3K	Cancer
	6	476**	*	*	9K	AIDS
	4	476**	*	*	5K	Cancer
EC2	7	476**	*	*	8K	Chlamydia
	8	476**	*	*	11K	Cancer
	9	476**	*	*	6k	AIDS
EC3	2	476**	*	*	7K	Syphilis
	1	476**	*	*	4K	Malaria
	3	476**	*	*	10K	AIDS

TABLE 2.5: Version of Table 2.1 where 0.5-closeness is applied to the Sensitive Attributes "Salary" and "Disease"

.

reasoning behind our selection.

Uniqueness. A record is considered to be unique if its equivalence class is of size 1. As explained in Section 2.3.1, such a record is at high risk of attribute and identity disclosure. To release a dataset while avoiding uniqueness, one could apply one of the privacy measures defined in Section 2.3. However, Zayatz argues that if a record is unique in the released sample but not in the original population, then it should not be considered unique (1991). To know the uniqueness of records in the original population, one either need access to it to compute it directly, or he has to estimate it. The first option is often not possible and is tedious as it needs to be updated regularly to reflect changes in the population. Numerous approaches have been proposed in literature for estimating the uniqueness in the original population, some of the most well known ones being Zayatz' model and Pitman's model (Zayatz, 1991;Pitman, 1996). These approaches assume that the population is drawn from a superpopulation by an appropriate distribution. For a thorough comparison and rule based approach to decide which model to use, the reader is encouraged to read the work of Dankar et al. (2012). It is important to note that, although these estimators do not need the original population, they do however need information such as the size of it and the sampling fraction.

Prosecutor Risk. A scenario in which an attacker is assumed to know that the record of an specific individual is contained in the dataset is referred to as a prosecutor model. Given that he knows that the individual is in the dataset, he is also able to infer in which EC he is contained. Therefore, the attacker's probability of correctly guessing which record of the EC is the individual is defined as

$$Prosecutor\ risk = P(Linkage|Membership) = \frac{1}{|EC|}$$

Journalist Risk. In the Journalist model, the attacker tries to re-identify any record in the dataset T, to prove that it is feasible, without knowing about any individual whether he is contained in T. Hence, the probability of successful re-identification is $P(Membership) \times P(Linkage|Membership)$. To achieve membership

disclosure, the attacker needs to have access to an identification dataset, denoted by Z, containing every individual also contained in T. Z needs to be generalized to the same level as T, at least conceptually. This results in having every EC in the target dataset correspond to an EC in Z that has at least as many records. Let EC_T and EC_Z be the ECs, in T and Z respectively, containing the individual I. Then the probability of re-identification of I is defined as follows:

$$\begin{aligned} P(re-identification) &= P(Membership) \times P(Linkage|Membership) \\ &= \frac{|EC_T|}{|EC_Z|} \times \frac{1}{|EC_T|} \\ &= \frac{1}{|EC_Z|} \end{aligned} \tag{2.1}$$

Given that the easiest re-identification would suffice, the worst case scenario is assumed where the intruder targets a record in the smallest EC_Z. Hence, the journalist risk is defined as $1/min(|EC_Z|)$.

Marketer Risk. Under the marketer model, the attacker tries to re-identify as many individuals as possible without knowing whether they are in the database. Hence, the risk measure in this case is the expected average number of re-identified individuals where Equation 2.1 is used for each (El Emam, 2010).

Average Equivalence Class Size Risk A common, yet simple, measure for re-identification risk is to use the average EC size (Li, Li, and Venkatasubramanian, 2007). It is related to the marketer risk and provides an upper bound for it. Indeed, every EC in the target database should have a corresponding EC in the identification database that is at least as big.

It is important to note that from the previously described risk measures, only the Prosecutor risk and the Average EC size risk do no need to make assumption about the underlying population nor have access to the identification database. This makes them well adapted to be used in the framework of our tool. Furthermore, it is interesting to remark that protecting against the prosecutor risk also protects against the journalist and the marketer risks (El Emam, 2013).

2.5 Utility Loss Measures

The attribute generalizations and the record suppressions, resulting from the privacy policies, inherently decrease the quality of the data. Therefore, it is important to measure this information loss in order to take it into account when increasing the privacy of the dataset. When releasing data, one has to balance between privacy and utility. In order to do so, multiple utility loss measures have been developed with differing degrees of complexity and for various scenarios and objectives.

Precision. This metric is based on the idea that the more an attribute is generalized, the more information is lost, and that if all attributes are generalized to the highest level then the data loses all its value. For a cell corresponding to attribute i and to record j, its value of distortion is defined as being the ratio between its level of generalization, $G_{i,j}$, and the total height of the generalization hierarchy for that

attribute, H_i. The precision of the released table T_R, having N_A number of attributes, is then 1 minus the sum of cell distortions normalized by the number of cells, as defined below:

$$precision(T_R) = 1 - \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{|T_R|} \frac{G_{i,j}}{H_i}}{|T_R| \times N_A}$$

Discernability. Another commonly used metric for measuring information loss, that of discernability, focuses solely on the size of the ECs and on the suppressed records (Bayardo and Agrawal, 2005). It assigns a penalty to every record that is proportional to the size of its EC. For suppressed records it assigns a penalty equal to the whole dataset, as such records are indistinguishable from every other record. This means that it does not take into account the level of generalization of attributes directly. For example, let G_1 be a version of the dataset that has one attribute generalized to the first level and let G_2 be a version of this same dataset that has another attribute generalized to level 4. If the generalization in G_1 results in larger ECs than G_2 then, under the discernability metric, G_1 is considered to have lost more information, regardless of the generalization level and of the height of the generalization hierarchy.

Non-uniform entropy. Building on top of the concept of entropy, the non-uniform version of it has been adopted in recent papers as a reference metric for evaluating the utility loss of a dataset (Dwork et al., 2006). From its predecessors it keeps the core concept that generalizing an attribute with many values results in a greater loss of utility than generalizing one with few values. For example, generalizing "Gender" from {male, female} to "*" provides less information loss to an attacker than generalizing "Age" from {0-100} to "*". The idea behind this is that, before knowing the value of the attribute, the intruder has a 50% chance of correctly guessing the gender, against a 1% chance of correctly guessing the age. Hence, knowing the age gives him more information than knowing the gender. Non-uniform entropy has the added feature of taking into account the distribution of an attribute. For example, generalizing the gender fo a dataset containing 1 male and 999 females should result in a much smaller information loss than for a dataset where the gender is uniformly distributed. This notion of unbalanced attribute distribution in captured by the non-uniform entropy.

As will be explained in Chapter 5, for our platform, we chose to use Precision over the two other measures because the intuition of it is most easily understood by the end user. This is in line with our vision for a tool in which the user can understand the impact that his de-identification has on the data, and thus on the utility loss. Therefore, for are more in-depth definition of the Discernability and the Non-uniform Entropy measures, we refer the reader to the works of Bayardo et al. and Dwork et al. respectively (Bayardo and Agrawal, 2005;Dwork et al., 2006).

Chapter 3

Planning

As explained in Section 1, some of the requirements, although minimally restrictive, are imposed by the SMOOTH project. However, we soon decided that a considerably more interesting tool could be developed. Therefore, in this Chapter I define more challenging objectives that make our tool innovative and stand out from the current state-of-the-art. I will also detail the timeline that I set for myself at the beginning of the implementation as well as describing how it changed over time.

3.1 Objectives

GR^{2}ASP aims to be a tool for analysing re-identification risk accessible to users with little knowledge of data privacy concepts. It should be easy to use, shielding the user from most of the inherent complexity by having him interact with an intuitive interface. The tool should incorporate the well established privacy policies defined in Section 2.3. Furthermore, measures of re-identification risk and utility loss, such as defined in Sections 2.4 and 2.5 respectively, should be shown to the user for him to understand the impact that transformations have on his data. Moreover, the user should be provided with a clear analysis of what causes the most risk in his dataset, through visualizations, as well as be provided with recommendations to transform his data most efficiently. We do not want to provide the user with an "optimal" solution that combines many small transformations and where the individual impact of each is lost, such as other similar tools do. On the contrary, we want him to understand the impact that each recommendation would have on the data, both in terms of re-identification risk and utility loss, so that he can make informed decisions on how to transform his data.

The tool should function both as a module of the SMOOTH platform and as a standalone application. In the first scenario it will receive the data and the types of the attributes as well as their generalization hierarchies. However, in the latter scenario, it will have to allow the user to upload data, overcome the lack of generalization hierarchies and make assumptions about the types of the attributes. Giving that there are only minor changes between the two versions, I will not distinguish between them except when explaining the specific differences.

For the implementation of GR^{2}ASP, I have defined the following principles which will guide the development of the platform.

Simplicity. The end-user of our platform is expected to be a layman with no strong knowledge of privacy concepts nor proficiency with complex software. Therefore, ease of use is one of the guiding principles throughout the development

of our tool. It is mentioned above that the user should be shielded from most of the complexity and be only exposed to concepts that are relatively easy to grasp. In line with this principle, we decided to build our tool as a dashboard, where most of the information is contained within one display. This allows the user to have the whole picture of what he can do with the platform at once, without having to go back and forth between displays and intricate menus. This ease of use comes at a cost, namely restricted flexibility and less "optimal" solutions. However, we argue that, in this scenario, simplicity outweighs both restrictions for the following reasons. First of all, we think that giving extensive flexibility to users that do not sufficiently understand the options available would lead them to either not chose or to randomly pick an option. This in turn, would probably not improve the results. Secondly, optimality of a solution depends greatly on the context, such as what the purpose of the output data is and which attributes are most important, and this is difficult to infer without receiving an active input from the user. Hence, the simplicity and intuitiveness of our tool should lead the user to preserve attributes that are important to him while generalizing the other ones, in order to achieve a solution that, although not optimal, would be very good for his purposes.

Parameterless. Extending on the above principle of simplicity, we want to build a tool in which the user will not have to define parameters. This does not mean that he will not have any choice to make, for he can choose which recommendations to apply. Rather, it means that he does not have to decide on which re-identification risk and utility loss measures to use, as these are decided by us. Furthermore, he does not have to define parameters for the privacy policies. For example, we define the value of k in k-Anonymity directly from the combination of risk and utility loss that the user wants to achieve. This definition overlaps with other principles defined in this section, however it is important to note that instead of having the user decide on k and afterwards see the risk and utility loss, our platform works the other way around. Therefore, we do not consider the risk and utility loss as a parameter but rather as guiding metrics used to decide which level of k to apply.

From Risk Analysis to Privacy Policy. Similar tools aiming at providing re-identification risk analysis, such as ARX , tend to function in a linear way, from defining the parameters to risk analysis, as can be seen in Figure 3.1. These tools do not make it easy for the user to make small changes and see their impact on the risk and utility loss metrics. On the contrary, most of the choices are made at the beginning and then the user only gets to see how they affect the measures after having processed the data. This requires him to have knowledge about privacy concepts in order to define the parameters efficiently beforehand. To overcome this, our tool should show the impact on risk and utility that each recommendation has, directly in the same interface. By doing so, we incorporate both metrics in the decision making of the user. He is shown recommendations and their corresponding measures and is able to decide for himself whether applying the recommendation is worth it. This results in a cyclic flow between receiving and applying recommendations, as is illustrated in Figure 3.2.

Explainable recommendations. The impact that transformations have on the re-identification risk and utility loss depends on which transformations have already been applied, that is, they are dependent on each other. Hence, let N_A be the number of attributes and let L_i be the number of generalization levels for the attribute i, then the number of possible combinations of transformations is given by the following

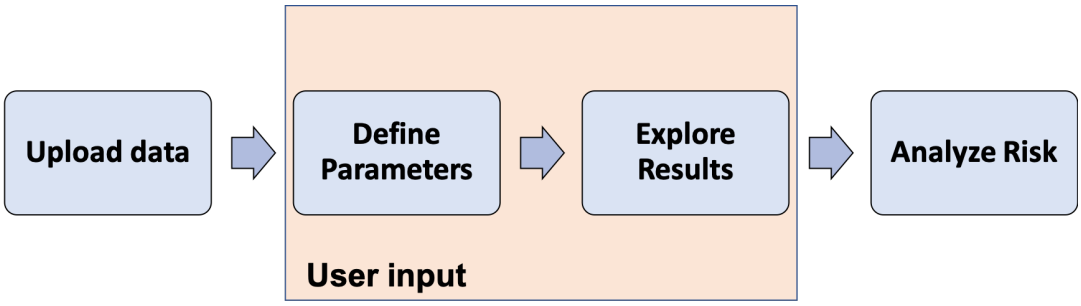

FIGURE 3.1: Human interaction and flow in ARX

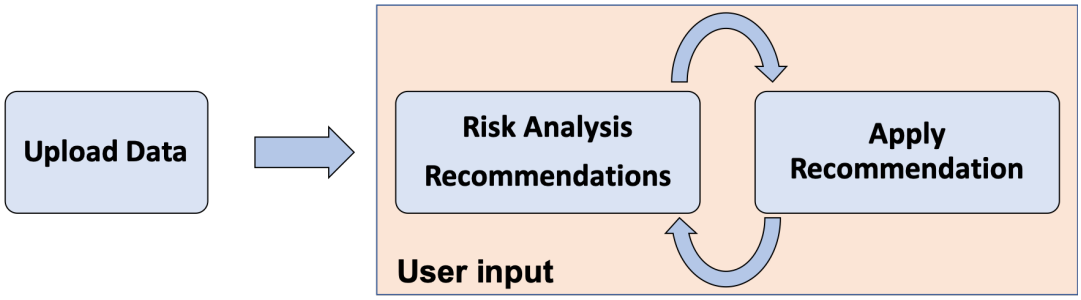

FIGURE 3.2: Human interaction and flow in GR^{2}ASP

formula:

$$combinations\ of\ transformations = \prod_{i=1}^{N_A} L_i$$

Even with only a few attributes each having a few levels, this number becomes large rapidly. For this reason, other tools such as ARX only show a limited amount of combinations. Our approach, however, is not to provide combinations of transformations directly, but rather to create them iteratively. Firstly, the user is shown transformations that affect only one attribute. Then, once he has applied one, he will be shown combinations of transformations that differ from the previously applied one by only one attribute, and so forth. This allows him to understand the impact that each individual transformation has, which we define as having explainable recommendations.

Human in the loop. The iterative interaction between the user and the platform is already mentioned above, however, being at the core of our tool, we want to stress the importance of it. Although the concept of "human in the loop" is not new, recent research has successfully been exploring ways to improve the outcomes of algorithms by involving a human in the decision making process (Li, 2017). One such example focuses on improving Machine Learning algorithms for health data, where rare events and small datasets often occur (Holzinger, 2016). While our scenario is not identical, we do think that it provides an interesting opportunity for improving the de-identification of the data by having a human in the loop. The importance of attributes is subjective on the context, and considering all of them to be at the same

level inherently diminishes the quality of the solution. Although tools like ARX allow the user to change the importance scale of each attribute, he has to do so without being able to know the impact that it will have on the de-identification process nor on the risk and utility. As mentioned above, this complicates his decision making. Furthermore, for certain privacy policies, the risk and utility measures rely on the generalization hierarchies to define the semantic closeness between values. If choosing the "optimal" solution based on these policies and measures, such as other tools do, one has to trust that they are able to accurately reflect the importance of the values and the attributes. However, this cannot be guaranteed, and is often not the case, as is explained in Section 2.3.3. In our platform, by letting the user iteratively choose which recommendation he wants to apply, we let him implicitly define the importance of each attribute and the quality of the generalizations. In our scenario, having a human in the loop has thus a double benefit. It improves the solutions while at the same time improving the user's understanding of what happens to the data and what causes the re-identification risk. Figures 3.1 and 3.2 illustrate the interaction from the user in ARX and in GR^{2}ASP respectively. In the first case, one can see that the user does not input information anymore once the result is obtained, whereas in the latter case, the user inputs information in a cyclic manner.

3.2 Timeline

Given that the implementation of GR^{2}ASP is done mainly individually by me with only intermittent help from colleagues at Eurecat, it did not lean itself well for a group project methodology such as Agile. However, the concept of incrementally adding features through sprints has been incorporated. Doing so allows me to get regular feedback, on a working product, from my supervisors, colleagues and partners in the SMOOTH project. After the literature review, feasibility evaluation and planning, the development was articulated around 3 main sprints.

Sprint 1. The milestone that is aimed at for this phase is to create a functioning dashboard that is able to give some measure of re-identification risk. We refer to the result of this sprint as the minimum viable product, in the sense that it would already satisfy the requirements imposed by the SMOOTH project, although, in a minimal way.

Sprint 2. In the second phase I implement that the user receives recommendations on how to transform his dataset, which I consider to be one the the main features that distinguishes our tool from similar ones. These recommendations, which can either be attribute generalizations or record suppressions, will require considerable coding in the back-end, as well as changes to the front-end to visualize them.

Sprint 3. The last phase has been added towards the end of the initial deadline of the thesis. With my supervisors from Eurecat, we decided that it would be worthwhile to postpone the deadline in order to improve and extend some of the features. Namely, we wanted to be able to handle datasets for which no generalization hierarchies is provided, speed up the platform and allow more time for other minor improvements. To overcome the lack of generalization hierarchies, I implement a method to generate them, both for numerical and categorical values. Speeding up

the platform relies mainly on tweaking the code, but also on changing the approach used for recommending record suppressions, as is explained in Chapter 5.

Chapter 4

System Overview

As mentioned above, one of the main guiding principles is that of simplicity. This led me to come up with a design that is presented as a dashboard where most of the information and options available to the user are seen at first glance. The only exception to this being the tab for record suppressions. When GR^{2}ASP functions as a standalone application, the user will, upon instantiating the tool, be prompted with a window requiring him to upload his dataset, as can be seen in Figure 4.1. After having done so, he will be shown the dashboard interface, which is identical to when the tool functions as a module of the SMOOTH platform. In all the screenshots of the platform, the dataset used is an open one, namely the "Adult" dataset of the UCI machine learning repository (Dua and Graff, 2017).

Interactivity. One of main the engaging aspects of the front-end is its interactivity, although this is obviously not clearly apparent through the static images displayed in this paper. As can be seen in every screenshot in this Chapter, above each metric visualization there is a "?" that the user can hover over with his mouse. Doing so will display a small box explaining the metric at hand. This is very useful for the user to help him understand the platform, while at the same time it allows us to de-clutter the tool from information that is not highly relevant. Furthermore, the user can also hover over the risk distribution to see the exact values. This allows him to get more accurate information while again helping towards having a sober interface. The user also interacts with the interface to apply recommendations, and the interface then updates smoothly to provide him with new information. For example, Figure 4.5 shows the *Current State* interface after two generalizations and 4-anonymity have been applied. Hence, we can see that less records are remaining, that the attribute *Occupations* has been suppressed and that the attribute *Workclass* has been generalized by one level.

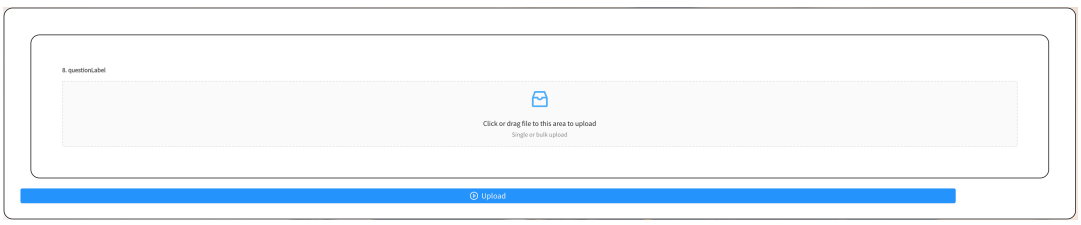

FIGURE 4.1: Upload interface

4.1 Current State of Data

Although the entire dashboard fits on a screen, for ease of visualization, in this paper we only show one half at the time. Figure 4.2 shows the upper half of the dashboard, which corresponds to the current state of the data as well as its risk and utility analysis. The left part shows the data in its current state, which means that if generalizations have been applied it will be reflected directly here, as can be seen in Figure 4.5. Records that have been suppressed will be removed from here. The right part is composed of the following elements.

Average Risk Gauge. As mentioned in Section 2.4, the average class size is a good and intuitive measure for the re-identification risk. The notion of hiding in a crowd is easily understandable by a layman user. However, the user might hesitate as to what is a good size of equivalence class to protect his dataset. Therefore we use the formula $\frac{1}{Avg\ |EC|} \times 100$ to give a value on a [0;100] scale with an associated green to red color scheme. Values below 30 mean that, on average, the ECs are at least of size three. This offers decent protection, as most record are indistinguishable from at least two other records, hence we associate it with a green-yellowish color. Values below 20 and 10 are linked with colours each a bit greener as they provide better protection. Values above 30 and below 50 are associated with 2 shades of orange increasingly dark. The reasoning for this is that they have, on average, between 2 and 3 records per EC, which offers some protection. Values above 50 have increasingly dark shades of red, as this means that most records are unique, which does not offer any protection.

Highest Risk. Taking the prosecutor risk of an individual in the smallest EC gives us the Highest Risk. Hence, it is defined as $\frac{1}{smallest\ EC}$, and is an intuitive measure for the worst case scenario. We multiple this measure by 100 for the purpose of uniformity. Furthermore, the colour scheme is also the same as described above. It makes sense that red should represent the worst case in which the smallest EC has only one record, as this one is not protected at all. The reasoning for values linked to orange and green follow the same reasoning as for the average risk.

Utility Loss Gauge, As explained in Section 2.4, we chose to use the Precision measure rather than the Discernability or the Non-uniform entropy ones. The reasoning behind it is that it is the most intuitive approach, while still providing a good measure of utility loss. For consistency, we use the same scale and color scheme as in the previous gauges. We think that losing more than half the information of the dataset is too much, and losing up to a third is reasonable. However, this is subjective, the re-identification risk being the priority, is is not trivial to define a threshold at which too much information has been lost.

Risk Distribution. This graph shows the percentage of records affected by the prosecutor risk. As is explained in the next Chapter, the Risk Distribution graph has considerably changed since I first implemented it. The reason for this being that it has proven to be more difficult for users to understand it. The final version, as can be seen in Figure 4.2, uses again a similar colour scheme as for the gauges in order to guide the user towards what should be a good distribution. The X-axis represents the different levels of prosecutor risk, while the Y-axis represents the percentage of records affected per level of risk. Lower risk values are better and are thus represented with green, while high risk values are represented with red. As will be

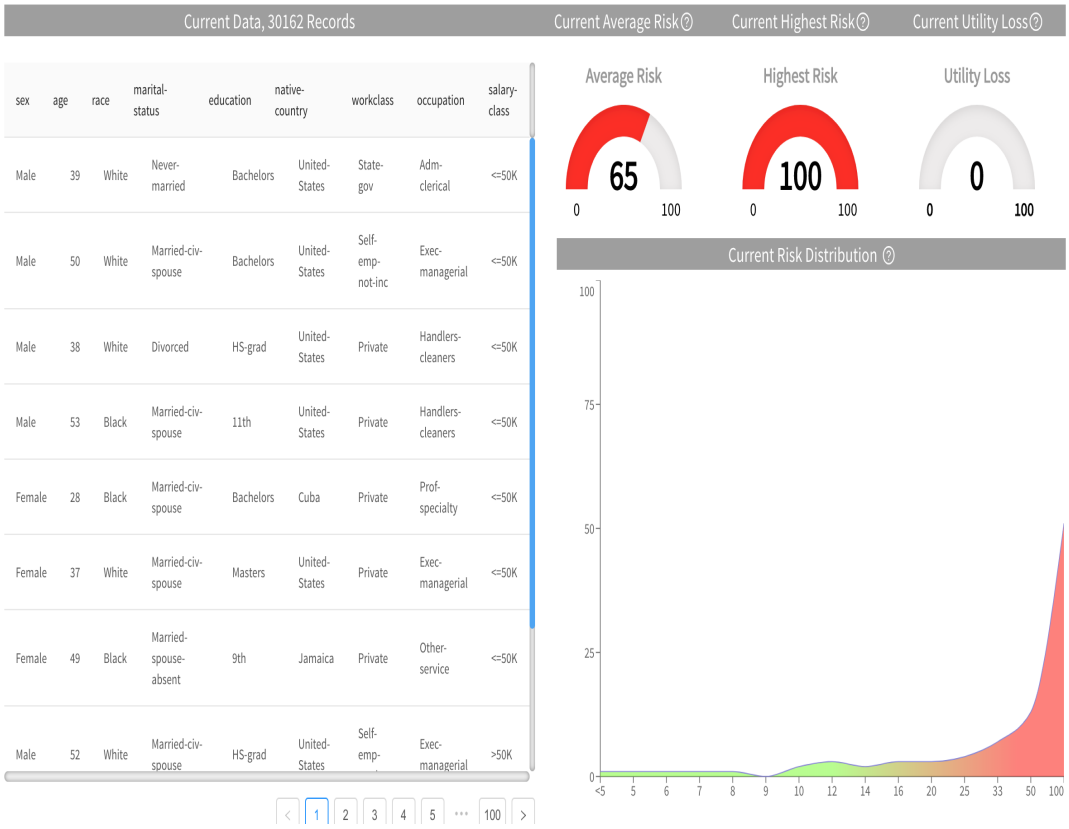

FIGURE 4.2: Interface showing the current data with its risk and utility visualizations before applying any recommendations

mentioned below, the highest value of k that the user can select is 20, which would make the smallest EC be of size 20. The prosecutor risk is given by $1/|EC|$, and thus an EC of size 20 corresponds to a prosecutor risk of 0.05, which we have defined as being the smallest level of risk represented by the graph. However, we multiply the risk by 100 in order to keep the same scale of for the other metrics. The percentage of individuals whose risk is lower than 0.05, or 5 on the graph, is aggregated under "<5". This visualization is not the most intuitive one, however, through the colour scheme, I think that it can sill help the user to make decisions.

4.2 Recommendations

One of the key features of our platform is to provide recommendations to the user for minimizing the re-identification risk. Through separating generalization and suppression, and by focusing on one attribute at the time, they allow the user to draw insights as to where then re-identification risks come from in their dataset. They are applied in an iterative manner and their impact is directly shown on the data table in the upper part of the dashboard. Therefore, it is important that the user is able to see the recommendations and the data simultaneously at all times, which is the case. The two kinds of recommendations, namely attribute generalizations and record suppressions, have each their own tab.

Attribute Generalizations. The first tab corresponds to recommendations based on generalizing the attributes, as is shown in Figure 4.3. This tab is divided in two

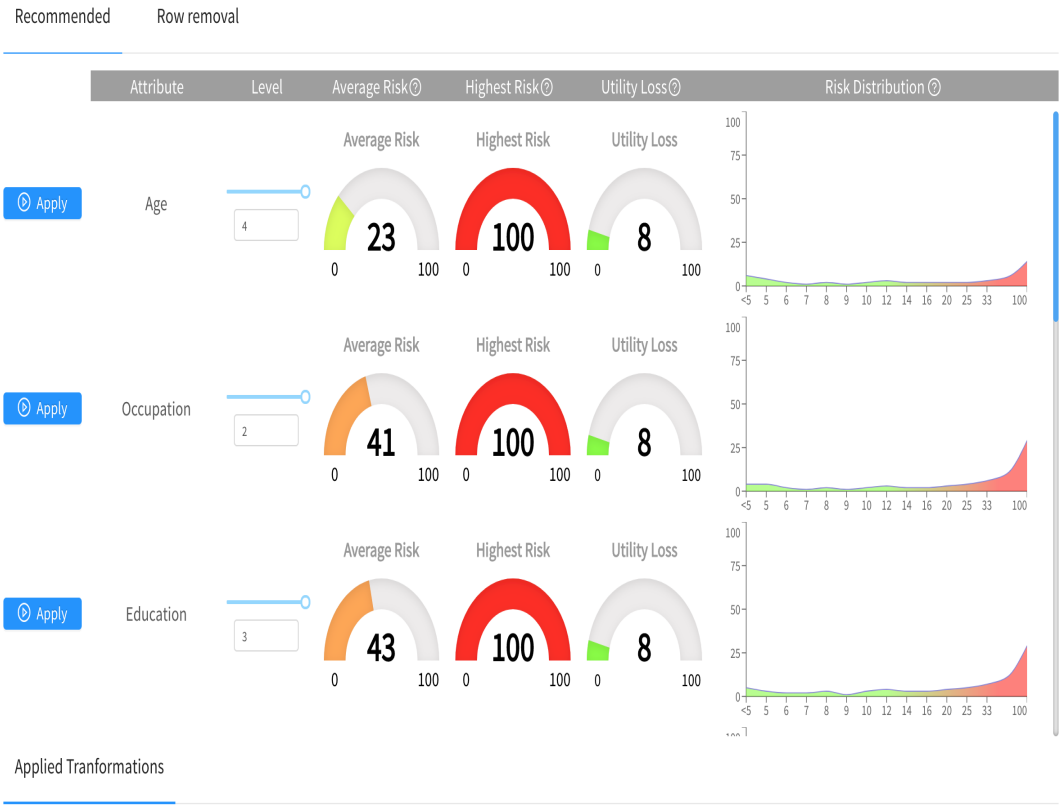

FIGURE 4.3: Interface showing attribute generalization recommendations before applying any recommendation

parts. The upper part shows a scroll-able list of recommendations that the user can apply, while the bottom part shows the ones applied already and allows him to undo them. Each attribute has its own row, with its name, a slider and a selection box with the possible generalization levels and the same gauges as in the current state part of the dashboard. When the attribute has only one generalization level, the slider and selection box will not be shown. However, when the attribute has several generalization levels, the user can choose different levels and directly see the impact on the corresponding visualizations. Furthermore, the recommendations are ordered from best to worst, where the best is considered to be the one that has the smallest sum of the three gauges.

Once an recommendation is applied by the user, it will be shown in the bottom part of this tab under "Applied Transformations", where he can choose to undo the transformation, as can be seen in Figure 4.6. As soon as a recommendation is applied, the whole dashboard updates. This implies that the current state is updated with its corresponding gauges, and that new recommendations are shown that take into account the previously applied ones.

Record Suppression The second tab of the bottom part of the dashboard corresponds to the record suppression recommendation, and is shown in Figure 4.4. The recommendation is similar to the attribute generalization ones in the sense that is has the same gauges, slider and selection box, and apply button. However, the value selected correspond to the k value of the k-Anonymity privacy policy. If the user wants a specific value for k, he can directly select it. Otherwise, the gauges will

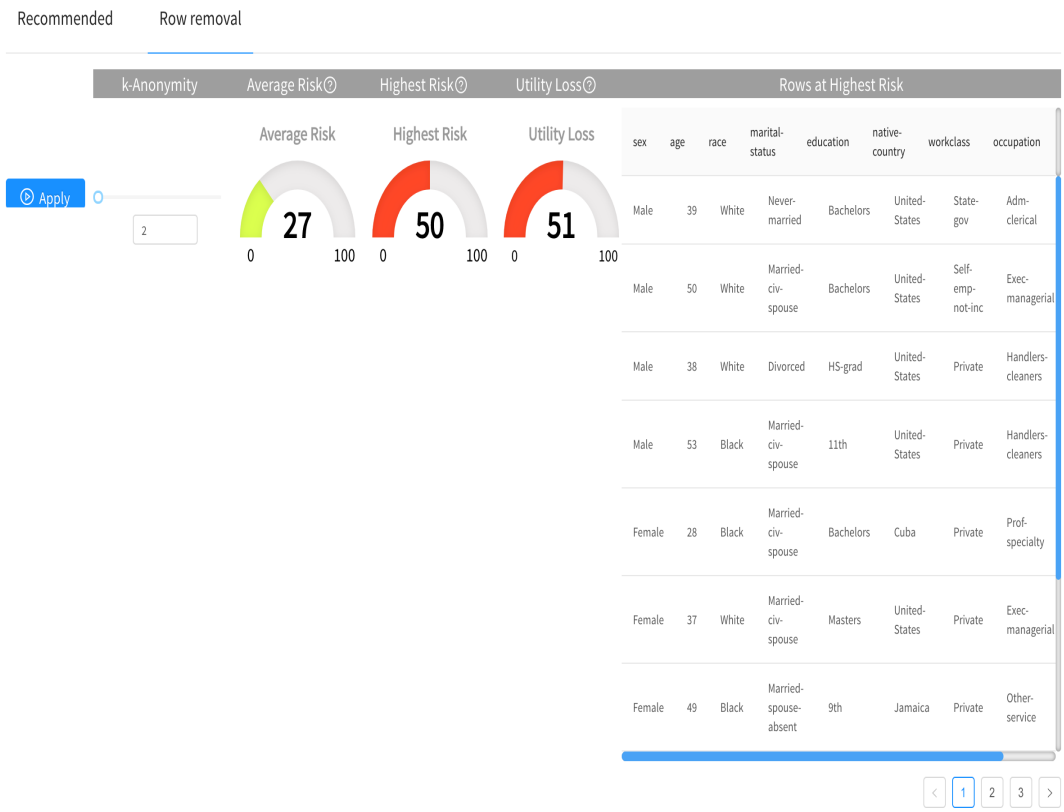

FIGURE 4.4: Interface showing record suppression recommendations before applying any recommendations

guide him to a level where the balance between re-identification risk and utility loss suits his purposes best. Initially, the user can select values ranging from 2 to 20. This range has be chosen so as to encompass the most common k values found in literature (LeFevre, DeWitt, and Ramakrishnan, 2005). Once the user has selected a value, the dashboard updates the current state and the recommendations, and the values for k will range from the selected value + 1 until 20. Furthermore, the data table on the right shows the records that are in an EC of the smallest size. These records are thus the ones at highest risk of an de-identification attack. This allows the user to understand where the risk comes from, through directly seeing the individuals that have the least protection.

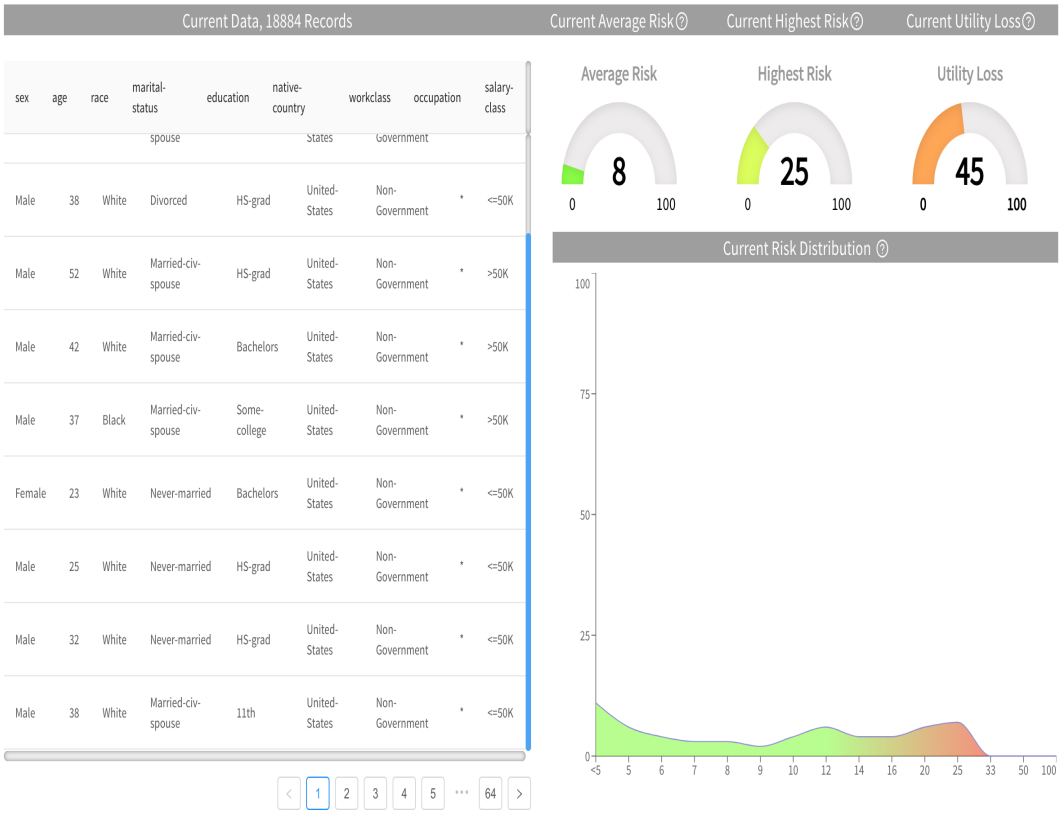

FIGURE 4.5: Interface showing record suppression recommendations before applying any recommendations

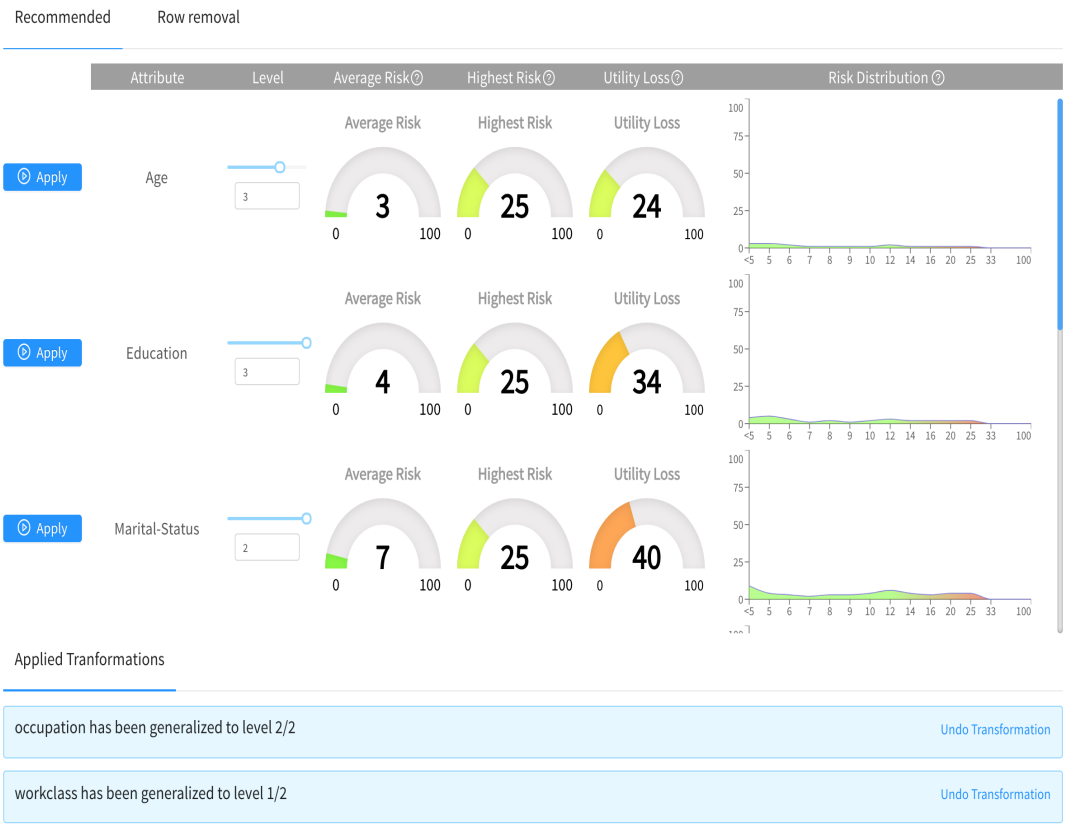

FIGURE 4.6: Interface showing record suppression recommendations before applying any recommendations

Chapter 5

Implementation

The back-end of GR^{2}ASP is built on top of the open source ARX library written in Java, as is described in Section 1.4. Initially, I forked the original repository on Github and implemented the methods that I needed wherever they were best suited in the ARX library. This often means that I would implement the methods within the class on which they operate, to be in line with the Object Oriented programming style. However, it soon became apparent that the whole platform could be programmed without needing to implement any methods inside the ARX classes and for a negligible sacrifice in speed and coding simplicity. This allows to import the ARX library rather than fork it, which in turn makes GR^{2}ASP much lighter in weight and allows to implement the updates of ARX more easily.

Although the back-end is entirely built by myself, the front-end has been made for a important extend by my colleague, Javier Cano. I have, however, made many changes to it once the general framework was implemented, as I will explain below. I will therefore not delve deeply into the implementation of the front-end, but rather focus on the back-end.

5.1 Back-end

The Unified Modeling Language (UML) diagram in Figure 5.1 shows the structure of GR^{2}ASP. The classes in green are implemented by myself, whereas the red ones are the classes from the ARX library which are directly used in our platform. I do not show all the classes involved in the functioning of the library, as it does not provide useful information in the context of this paper. A more complete UML diagram of ARX can be found in the related paper (Prasser et al., 2014).

The SmoothIO class holds all the logic implemented for GR^{2}ASP, and is instantiated through the SmoothIOController. In the SmoothIO class I define an ARXConfiguration which is passed to an instance of the ARXAnonymizer to obtain an ARXResult. These first two ARX classes are omitted from the diagram for clarity purposes. The first one defines the parameters, such as the privacy policies, while the second's purpose is to anonymize the data according to the configuration. SmoothIO also holds an instance of the Data class, which represents the original state of the data. The main purpose of ARXResult is to be the controller of the ARXLattice. This latter class contains all the possible anonymizations, that is, one per combination of attribute generalizations. The structure of the graph depends on the generalizations, therefore the neighbors of a node differ from it by one level of generalization for one attribute. Figure 5.2 shows a small sample of the lattice for the "adult" dataset, where the numbers represent the generalization levels of

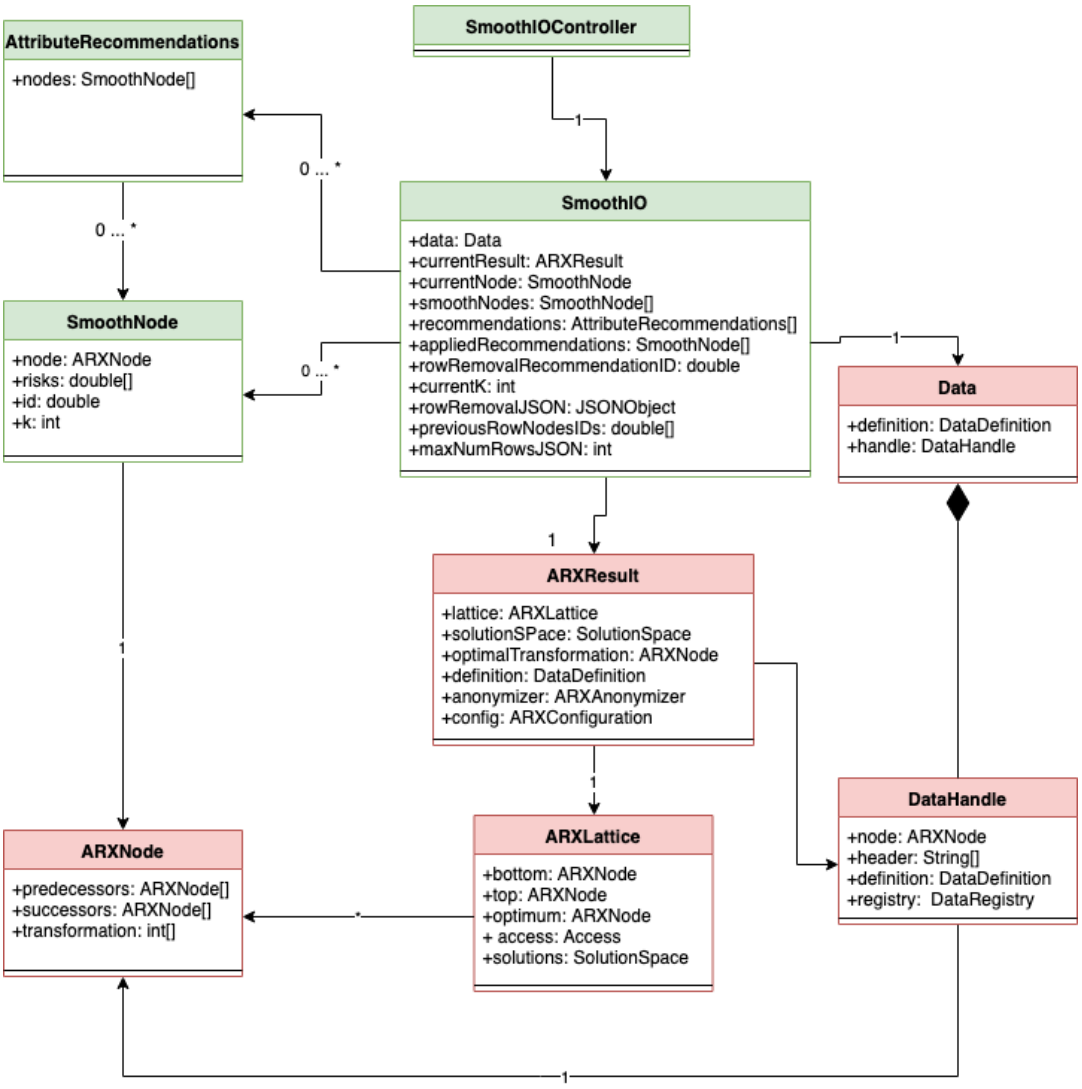

FIGURE 5.1: UML diagram of GR^{2}ASP

the attributes. ARXLattice is thus mainly a graph structure, where each node is an ARXNode, that provides access to the bottom, top, and optimum node. The optimum node represents the transformations that are considered the best given the utility measure, however our platform does not use this, as I explained in Section 3.1. The ARXNodes provide access to their predecessors and successors, as well as additional information such as the transformations applied to the data. A DataHandle, which represents the data given the applied transformations of the given node, is obtained by providing an ARXNode to the ARXResult. Each SmoothNode contains one ARXNode and is implemented in order to provide additional information, such as the risk and utility measures and a ID which is used to interact with the front-end. The AttributeRecommendations class is a data structure that contains nodes generalizing the a same attribute and that can be sorted based on the metrics of the nodes. Initially, SmoothIO creates an ARXResult based on 1-anonymity. This privacy policy does not impose any transformation, its purpose being only to create a result with its corresponding lattice. From this result we compute the risk and utility measures shown in Figure 4.2.

Attribute recommendations. It can be seen in Figure 4.3 that the interface

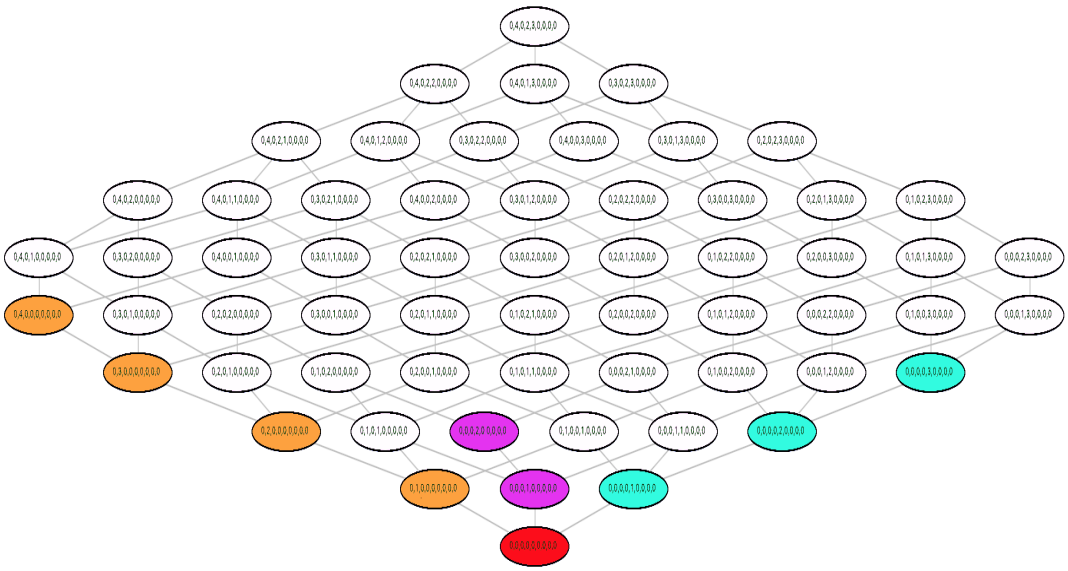

FIGURE 5.2: Sample of the ARXLattice of the "adult" dataset focusing only on 3 attributes. The colors illustrate the depth-first search of GR^{2}ASP starting at the bottom node.

presents the attribute recommendations as a list where, for each attribute, the user can select the level of generalization. This reflects the underlying structure that I have implemented, that is, as a list of lists of nodes. However in ARX, as explained above, the structure used is a graph. Therefore, I implemented a depth-first search of the graph based on the generalization level of one attribute at the time. Figure 5.2 illustrates this search for three different attributes, the nodes generalizing the other attributes having been left out due to the size constraint. In this figure, it can be seen that the search starts at the bottom node of the graph and each color represents the path for one attribute. The nodes are then stored in lists, namely *AttributeRecommendations* objects, per attribute that is being generalized. When the user applies an attribute generalization in the front-end, the ID of the node is sent to the back-end to identify which node will become the new *currentNode*. Providing this node to the ARXResult gives a new DataHandle that reflects the generalization of the attribute and from which the new metrics are computed, both of which can be seen in Figure 4.2. As mentioned in Chapter 4, once an attribute generalization has been applied, all the recommendations need to be updated to incorporate this change by fixing the level of the given attribute. In order to do so, I define the *currentNode* as starting position for the graph search. An example of the depth-first search when the second attribute is fixed to the fourth level can be seen in Figure 5.3. The process of computing new attribute recommendations keeps on repeating as long as the user applies recommendations. When an attribute generalization is undone by the user, a similar process happens but in a top to bottom direction rather than a bottom to top direction as previously. When the user removes any of the applied recommendations, the ID allows to identify of which attribute the generalization level should be set back to 0. To do so, I define the *currentNode* as starting point for the depth-first search and the search explores the predecessors decreasing the generalization level of the given attribute.

Record Suppression Recommendations. The approach used for creating the

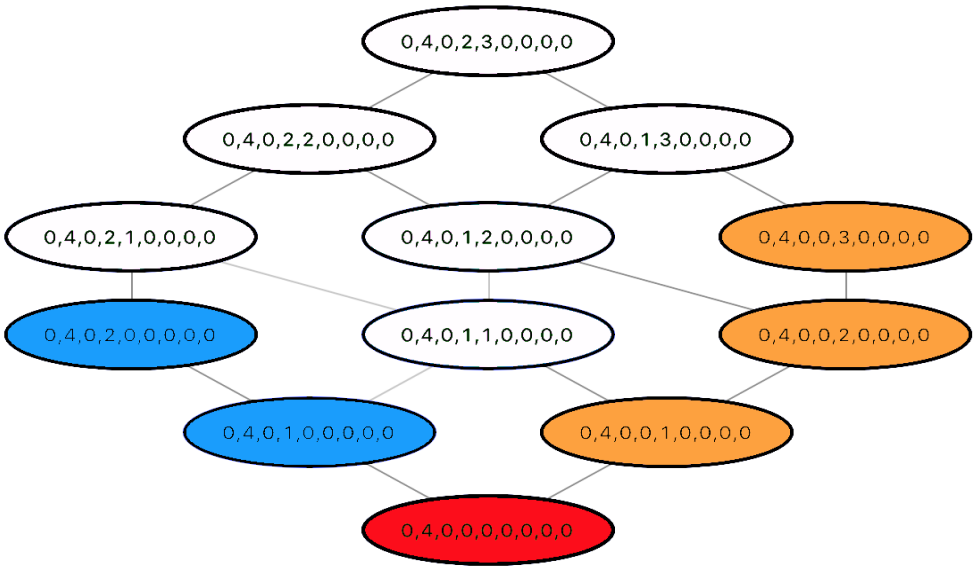

FIGURE 5.3: Illustration of the depth-first search of GR^{2}ASP starting at the node with generalizations [0,4,0,0,0,0,0,0,0]

record suppression recommendations has considerably changed throughout the sprints. I will describe the approach implemented in the final version, and explain in Section 5.3 how it has evolved. The current version handles QIs as well as SAs, however I will first develop the approach for a dataset containing only QIs and then extend the explanation to the case with SAs.

The record suppression recommendations are based on k-anonymity. The user select a value of k between 2 and 20, and the corresponding re-identification risk and utility measures will be shown to him, as can be seen in Figure 4.4. However, whereas for the attribute recommendations these values could be gotten from the ARXResult directly, for record suppression this is not feasible, because running k-anonymity with multiple different values takes considerably too long. Nevertheless, being able to get the number of ECs per size in the current result allows for fairly easily computing these metrics. Let N_i be the number of ECs of size i, let I be the size of the biggest EC and let k be the value of k-anonymity, we then obtain the following formulas:

$$Average\ risk(k) = 1/(\frac{\sum_{i=k}^{I} N_i \times i}{\sum_{i=k}^{I} N_i}) \tag{5.1}$$

$$Highest\ risk(k) = \frac{1}{k} \tag{5.2}$$

$$Utility\ loss(k) = N_{previous\_k(k)} \times (previous\_k(k)) \tag{5.3}$$

Where we define $previous\_k(k)$ as being size of the greatest EC's size smaller than k. In general, this will equal to $k-1$, however I defined it this way to hold for the following scenario. For some values of k there might not be an EC of that size. When this is the case, the metrics shown to the user will be the ones from the next smallest value of k for which a corresponding EC is present in the dataset. The reasoning behind this that it has no purpose to achieve, through record suppression only, a level of k-anonymity for which there is no EC of the corresponding size. To do so, records from the next smallest EC would have to be suppressed, which decreases

the utility and increases the re-identification risk for no reason.

When one or more SAs are contained in the dataset, privacy policies protecting the sensitive attributes, such as defined in Sections 2.3.2 and 2.3.3, have to be used. However, applying these policies happens behind the scenes and the user is not involved in defining their parameters, not even indirectly as is the case for k-anonymity. The reasons for this are twofold, namely the simplicity objective and the lack of use cases. Allowing the user to indirectly set the parameters for l-diversity and t-closeness would require additional risk metrics that focus on the SAs. This would make the tool considerably more complex for only a slight increase in functionality. Furthermore, we do not expect this tool to be often used with SAs, mainly because, with the GDPR, few companies are allowed to process sensitive information. We include SA protecting policies for completeness sake, however, the primary focus of GR^{2}ASP remains to protect against identity disclosure rather than against attribute disclosure. For selecting which of the two policies to apply, our platform follows a simple rule based approach. If the SA is numeric, t-closeness is applied, for it is well suited for such cases and offers additional protection. When the SA is categorical, we have two options. Either the generalization hierarchy of the SA is provided, in which case t-closeness is used. Else, when the generalization hierarchy is not available, l-diversity is applied. For the reason that both policies tend to considerably decrease the utility of the data, and that the argumentation for high levels of both policies relies on strong assumptions about the knowledge of the attacker, the values that I set for both are fairly low in terms protection, namely $l = 2$ and $t = 0.5$.

It is important to note that when using privacy policies for SAs, it is not possible to accurately predict the metrics before applying the policy, as is done for k-anonymity. The higher complexity of l-diversity and t-closeness, combined with their interaction with k-anonymity make it impossible to obtain accurate estimates of the metrics, and considerably difficult to obtain even fairly good estimates. It is obvious that given a certain level of k-anonymity, adding one of the two other polices would decrease the risk and increase the utility loss of the dataset, however to what extend cannot be accurately predicted. Having long pondered on the problem, I decided to tackle it in the following manner. Regardless of the number of policies applied, the metrics will always be computed based solely on the k-anonymity values. My reasoning for this is that, any estimate that I could think of would be inaccurate and that, as mentioned above, I expect this tool to be rarely used with SAs.

Standalone Application. As mentioned in Section 3.1, GR^{2}ASP can function either within the framework of the Smooth platform, or as a standalone tool. However, in the latter case, the tool does not get the data, the attribute types nor the generalization hierarchies from the platform, and thus workaround methods have to be implemented.

Data Upload. As is shown in Figure 4.1, I have implemented, both in the back-end as in the front-end, that when the tool is functioning outside of the Smooth platform, upon starting the application it prompt the user to upload their data. In the back-end, I implemented this through a post request to the rest-API from the front-end. Whereas for the front-end I implemented the upload interface as well as the logic for sending the request. Furthermore, given that the attribute types are not defined by the user, all are considered to be QIs. As mentioned above, we do

not expect this tool to be often used with SAs, especially not when running as a standalone application.

Attribute Generalization Hierarchy. Given that, in the standalone case, GR^{2}ASP does not get the generalization hierarchies, I implemented a technique to automatically create them. For numerical attributes, the values are put into bins to form hierarchies similar to the ones found in literature. To do so, first the level of the hierarchy is defined. Let V be the set of values for that attribute in the dataset, the level L for its generalization hierarchy is then defined as follows:

$$L = min(\lfloor log_2(\lceil range(V) \rceil) \rfloor, 4) \tag{5.4}$$

The arbitrary upper bound of 4 comes from it being commonly used in literature as the number of levels for generalization hierarchies of numerical values (Li, Li, and Venkatasubramanian, 2007). Given that we group two bins together to form one bin of the level above it, we can define the number of bins for the first level as:

$$NB_1 = 2^{L-1} \tag{5.5}$$

The size of these same bins is then defined as:

$$SB_1 = \left\lceil \frac{\lceil range(V) \rceil}{NB_1} \right\rceil \tag{5.6}$$

For each bin of the second level, two bins of the first level are merged, starting with the smallest ones. This process is repeated for the third level and the last level always represents the suppression of the attribute, which is denoted by "*". As mentioned in Section 2.3.1, numerical attributes such as *ZIP code* are generalized by replacing the last digits with "*", which I argued is equivalent to putting into bins of $size = 10^{level}$, where level represents the generalization level. I posit however, that putting the values into bins of different sizes, such as is the case with this approach does not, or minimally, impact the quality of the generalizations. Lastly, by rounding up the range in Equations 5.4 and 5.6, this approach inherently also handles decimal values.

For categorical attributes, it is not trivial to reproduce the generalization hierarchies found in the literature, such as the ones received from the Smooth platform. To do so would require understanding the semantic meaning of the values, through Natural Language Processing, and be able to aggregate them in a generalization group that is coherent with the context of all the values. Even so, it remains subjective, and many hierarchies are flawed in the context of data privacy. For example, grouping countries on a geographic basis offers little protection against a region based bias. Implementing such techniques to create generalization hierarchies is outside the scope of this paper. Therefore, I developed a simpler approach that performs relatively well. The intuitive idea is akin to the one of k-anonymity, namely to hide in a crowd and thus, to group the values into sets making it more difficult for an attacker link an attribute value with a record. To do so, I start by defining the number of levels similarly as for the numeric values but where the range is replaced with the cardinality. Let V again represent the set of values, I define the equation to define the number of levels:

$$L = min(\lfloor log_2(\lceil |V| \rceil) \rfloor, 4) \tag{5.7}$$

The amount of information lost by generalizing with this approach is greater than for the numeric values and than when using the semantic generalization hierarchies. The reason for this is that it is more difficult to draw insights about a record of which an attribute is a set of values that are not related semantically, compared to the case where the attribute is a set of values related, or the name of what defines this set. To limit this drawback, I hard code the number of values in the first levels to be as small as possible, rather than computing this number based on the total number of levels. This means that when the cardinality of the set of all values of an attribute is pair, the sets of the first level will each contain two values, whereas if it is odd, one set will contain three values. The frequency of the values is used to compose the sets. When there is a pair number of unique values, I put the least and most frequent value in the same set for the first level. Then, the second set will be made up of the second least and second most frequent values, and so forth for the other sets. When the cardinality of the set of all values is odd, I put the least and second least frequent values together with the most frequent value in the first set. Then the next sets follow the logic of the pair scenario. The procedure for creating the sets of the subsequent levels follows the one from the first level, where the values are replaced with the sets of the level below. The last level represents again the suppression of the attribute, and is denoted by "*".

5.2 Front-end

A considerable part of the front-end has been implemented by my colleague from Eurecat, Javier Cano. Therefore, I do not delve far into the details of the development, but rather I briefly explain the general framework, and I elaborate about the interaction with the back-end and about the changes implemented by myself.

To facilitate linking the back-end with the front-end, I decided to only use one HTTP request, namely a GET request with an optional ID parameter. Initially, the front-end sends the given request, without the optional parameter, to receive the JSON with the current state of the data and the recommendations. Each of these contains the data necessary to visualize it, as well as an unique ID for each level of each attribute generalization and each level of the record suppression recommendation. When the user applies a generalization at a specific level, the ID is sent to the back-end which retrieves the corresponding node, applies the generalization and sends a new JSON to the front-end containing the recommendation in the field *appliedRecommendations*. When a user undoes an applied recommendation, its ID allows the back-end to set the attribute back to level 0 or to set k back to 1, as is explained above, and then sends back a new JSON. Using only one request for all purposes made the coding slightly more complex in the back-end, however it allowed to easily tryout and implement changes without having to coordinate too often with the front-end development. As mentioned above and as will be detailed in Section 5.4, the record suppression recommendations have radically changed throughout the development. Therefore, I fully implemented the current version of the tab corresponding to these recommendations. Furthermore, I also implemented the data upload button as well as its functionality. And lastly, I made many small changes throughout the interface, such as making the data scrollable, making buttons load when waiting for the JSON, making the corresponding subsection disappear when no recommendation is available anymore, and a few other changes. I would like

to point out that my involvement in the front-end is mainly motivated by my desire to fully understand the whole tool, and be able to keep on making any kind of improvements myself in the future.

5.3 Improvements

Throughout the implementation of GR^{2}ASP, numerous changes have been made and certain paths have been explored that turned out to be unsuccessful for our purposes. These variations are derived either from my desire to make the tool faster and more robust or from feedback from colleagues and users. I do not elaborate on all the small changes, for these are inherent to developing a tool, but rather I will focus on those that have considerably impacted the final result.

Performance optimization. Many of the improvements that I made during the development are coding related, such as storing objects or values that are computed several times, using better data structures and taking more advantage of Java's Object Oriented way of programming. However, these changes are too numerous and detailing them brings little value to the reader. The change that improved the most the usability of our tool is the way that the record suppression recommendations are generated. Initially, I implemented these recommendations such that the user can only increase the value of k by one level at the time. The reason for this is that, to provide this recommendation, the tool would generate a new *ARXResult* with k being one level higher than the current one. This has as benefit that the metrics related to it can be directly obtained from the new result, and are thus exact. The drawback, however, is that this approach is considerably slow, especially when the user wants to achieve a high level of k-anonymity, either because he is familiar with the privacy or as a result of iteratively decreasing the risk to reach the desired level. Therefore, I decided to compute the metrics for k-anonymity without obtaining a new *ARXResult*, as is described in Section 5.1. However, this means that the metrics cannot be accurately displayed when l-diversity or t-closeness are applied. Nevertheless I think that the great increase in speed and usability is worth the trade-off in accuracy, especially given that I do not expect this tool to be regularly used with SAs.

User feedback. Given that it was not possible to involve the end users in all steps of the development, I could not follow a user-centered design process. Therefore, I designed the initial interface, with feedback from my supervisors. However, the target end users have little to no data privacy knowledge, and it was thus fairly trivial to find several individuals in my entourage that fitted this description and could test the tool. To do so, I watched them de-identify the "Adult" dataset to a level that provided, according to them, a good balance between risk and utility, and I asked them for feedback on their experience with the tool. This process proved to be helpful for identifying which parts were intuitively understood and which were not. Overall, the users managed to move around the interface and fulfill their task relatively easily, the only difficulty deriving from the metrics used and their visualizations. Figure 5.4 displays the metrics of the version of GR^{2}ASP with which the testers interacted. These are the same as for the Current State of the data and for the Record Suppression recommendations but in a different arrangement.

The first observation that became directly apparent is that the users did not understand nor use the Risk Distribution graph, therefore I changed it to the version

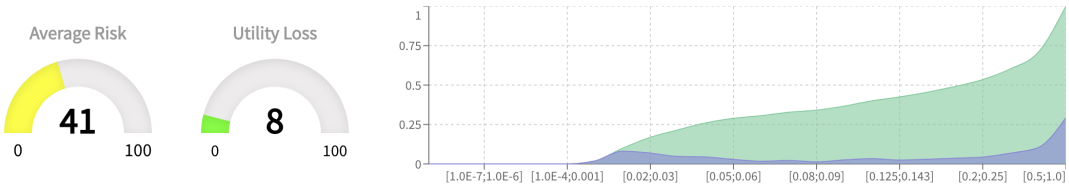

FIGURE 5.4: Average Risk, Utility Loss and Risk Distribution metrics of the attribute recommendations in a non-final version of GR^{2}ASP

shown in Figure 4.2. This graph imitates closely the one available in ARX, which represents one of its main risk visualizations, and I thus wrongly assumed that it would be intuitive for the users. A detailed explanation of it is omitted from this paper, as a new version as been implemented, nevertheless for completeness sake I describe it briefly. The X-axis represents the prosecutor risk put into bins of increasing size, and the Y-axis represents the proportion of records at risk. The blue graph displays the proportion of users per risk bin, whereas the green graph shows the cumulative distribution of the risk. Hence, the intuition should be to try to move the area of the cumulative distribution as much to the left as possible, for this would mean that more records are at a lower risk. Meanwhile, the blue graphs displays how the risk is distributed over the records, allowing the user to see what the worst prosecutor risk is and how many people are affected by it.

Secondly, I noticed that the users would generally stop applying recommendations slightly too early and thus not protect the dataset well enough from re-identification risk. Therefore, I changed the color scheme of the gauges to make it apparent that the level of risk at which the users would generally stop is not sufficient. The original green to red scale displays the values between 30 and 50 in shades of yellow. However, I shifted the starting point of orange downwards to start at 30 in order to reflect that from there to 50 provides just sufficient protection rather than good protection.

The last observation was that users would often not apply any record suppression recommendation at all. Their reason for this is that generally the attribute generalization recommendations provide a considerably better ratio between Average Risk and Utility Loss, hence the users would, quite logically, not suppress records. However, suppressing records is often the only way to efficiently protect against the prosecutor scenario, that is, to increase the size of the smallest EC. Generalizing attributes works best for decreasing the average risk, but for the highest risk suppressing records to achieve k-anonymity is essential. Therefore, this observation highlighted the flaw in not displaying the Highest Risk. Which I subsequently fixed.

Implementing the changes resulting from these observations has led to displaying the metrics for the attribute recommendations as is shown in Figure 5.5, which is again the same as for the current state of the data and for the record suppressions. I then repeated the experiment with some of the same users and some new ones. Although the experiments are not very rigorous, the users now seemed to manipulate GR^{2}ASP more fluidly and also reported that they easily understood all of its components.

FIGURE 5.5: Average Risk, Highest Risk, Utility Loss, and Risk Distribution metrics of the attribute recommendations in the final version of GR^{2}ASP

5.4 Development timeline

In Section 3.2, I explain the planning of the development, which I have tried to follow. To summarize briefly, Sprint 1 focuses on creating a minimal functioning product, in Sprint 2 I implement the feature which make it stand out, namely the recommendations, and in Sprint 3 I focus on extra features, such as hierarchy creation, and improving the overall tool.

Sprint 1 and 2 ended up being slightly merged together for several reasons. Firstly, it was not clear initially whether a colleague could help with the front-end and to which extend. Once this was decided upon, it took longer that planned to link the back-end to the front-end and several ideas for the design were tried out which also took time. Given that it would have little purpose to have a functioning minimal version without front-end, I decided to blend together the back-end development of both Sprints. Furthermore, once it was decided to add a third Sprint, and thus to extend the deadline of the project, I allocated more time to finish the second Sprint. During the last Sprint, and while nearing the final deadline, I had to decide to wrap up the project for it to be a fully functioning platform, and to allocate time to write this paper. However, this is not as trivial as it sounds, for having spent considerable time on this project and enjoying it more as it becomes more complete, I am continually tempted to add features. These ideas not yet implemented will therefore only be mentioned as future improvements.

Chapter 6

Conclusion

In this paper I have detailed the development of a tool fulfilling the requirements defined by the Smooth Project, namely to create an intuitive re-identification risk analysis platform. However, I have also shown that GR^{2}ASP goes far beyond the initial objective. Indeed, our tool guides the user, using explainable recommendations, through the process of analysing the re-identification risk and de-identifying his dataset. The human-in-the-loop approach, combined with the recommendations, is highly intuitive and provides great insight as to what causes the risk in the user's dataset. Furthermore, the user is shown clear visualizations which allow him to choose the right balance between re-identification risk and utility loss that he requires. The intuition inherent to the straightforward recommendations allows the user to understand and be fully in control of the transformations applied to his dataset. Although interacting with relatively simple concepts and measures, the user is indirectly applying more complex data policies such as k-anonymity, l-diversity and t-closeness without needing advanced knowledge of data privacy concepts. The combination of these novel features make our tool stand out from the current state-of-the-art, given the problem at hand. Although having mainly be developed with the aim of being part of the Smooth Platform, GR^{2}ASP can also function as a fully standalone application. To do so, I have implemented a data upload interface, as well as methods for automatically creating attribute generalization hierarchies, which is again a feature not found in similar tools.

The main stakeholders involved in the development of GR^{2}ASP and of the SMOOTH platform are pleased with the final result, and it is expected that it will be frequently used once the platform goes online. Given the real life application of this tool, as well as the fact that the deadline for the SMOOTH project is still relatively far ahead, I expect to keep on improving the tool in the near future. If time permits, I would like to provide the user with a clear textual explanation of the re-identification risk factors in his dataset. Lastly, should someone else be responsible for future development of GR^{2}ASP, I have extensively documented the code, and one should relatively easily be able to pick up from the current state.

Appendix A

Appendix A

FIGURE A.1: Gantt chart of development

Bibliography

Bayardo, Roberto J and Rakesh Agrawal (2005). "Data privacy through optimal k-anonymization". In: *21st International conference on data engineering (ICDE'05)*. IEEE, pp. 217–228.

Bélanger, France and Robert E Crossler (2011). "Privacy in the digital age: a review of information privacy research in information systems". In: *MIS quarterly* 35.4, pp. 1017–1042.

Berg, Harry Van den (2008). "Reanalyzing qualitative interviews from different angles: The risk of decontextualization and other problems of sharing qualitative data". In: *Historical Social Research/Historische Sozialforschung*, pp. 179–192.

Bergeat, Maxime et al. (2014). "A french anonymization experiment with health data". In:

Bild, Raffael, Klaus A Kuhn, and Fabian Prasser (2018). "Safepub: A truthful data anonymization algorithm with strong privacy guarantees". In: *Proceedings on Privacy Enhancing Technologies* 2018.1, pp. 67–87.

Bingisser, G Martin (2008). "Data Privacy and Breach Reporting: Compliance with Various State Laws". In: *Washington Journal of Law, Technology & Arts* 4.3, p. 9.

Campan, Alina and Traian Marius Truta (2008). "Data and structural k-anonymity in social networks". In: *International Workshop on Privacy, Security, and Trust in KDD*. Springer, pp. 33–54.

Christen, Peter (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Dankar, Fida Kamal et al. (2012). "Estimating the re-identification risk of clinical data sets". In: *BMC medical informatics and decision making* 12.1, p. 66.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Dwork, Cynthia et al. (2006). "Our data, ourselves: Privacy via distributed noise generation". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, pp. 486–503.

El Emam, Khaled (2010). "Risk-based de-identification of health data". In: *IEEE Security & Privacy* 8.3, pp. 64–67.

— (2013). *Guide to the de-identification of personal health information*. Auerbach Publications.

Gedik, Bugra and Ling Liu (2007). "Protecting location privacy with personalized k-anonymity: Architecture and algorithms". In: *IEEE Transactions on Mobile Computing* 7.1, pp. 1–18.

Ghinita, Gabriel, Panos Kalnis, and Yufei Tao (2010). "Anonymous publication of sensitive transactional data". In: *IEEE Transactions on Knowledge and Data Engineering* 23.2, pp. 161–174.

Gkoulalas-Divanis, Aris and Grigorios Loukides (2015). *Medical data privacy handbook*. Springer.

Gressin, Seena (2017). "The equifax data breach: What to do". In: *US Federal Trade Commission, as viewed Oct* 1.

He, Yeye and Jeffrey F Naughton (2009). "Anonymization of set-valued data via top-down, local generalization". In: *Proceedings of the VLDB Endowment* 2.1, pp. 934–945.

Heeney, Catherine et al. (2011). "Assessing the privacy risks of data sharing in genomics". In: *Public health genomics* 14.1, pp. 17–25.

Holzinger, Andreas (2016). "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" In: *Brain Informatics* 3.2, pp. 119–131.

Jain, Priyank, Manasi Gyanchandani, and Nilay Khare (2016). "Big data privacy: a technological perspective and review". In: *Journal of Big Data* 3.1, p. 25.

Kondylakis, Haridimos et al. (2018). "Implementing a data management infrastructure for big healthcare data". In: *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, pp. 361–364.

LeFevre, Kristen, David J DeWitt, and Raghu Ramakrishnan (2005). "Incognito: Efficient full-domain k-anonymity". In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, pp. 49–60.

LeFevre, Kristen, David J DeWitt, Raghu Ramakrishnan, et al. (2006). "Mondrian multidimensional k-anonymity." In: *ICDE*. Vol. 6, p. 25.

Li, Guoliang (2017). "Human-in-the-loop data integration". In: *Proceedings of the VLDB Endowment* 10.12, pp. 2006–2017.

Li, Ninghui, Tiancheng Li, and Suresh Venkatasubramanian (2007). "t-closeness: Privacy beyond k-anonymity and l-diversity". In: *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, pp. 106–115.

Lukács, Edit et al. (2005). "The economic role of SMEs in world economy, especially in Europe". In: *European integration studies* 4.1, pp. 3–12.

Machanavajjhala, Ashwin et al. (2006). "l-diversity: Privacy beyond k-anonymity". In: *22nd International Conference on Data Engineering (ICDE'06)*. IEEE, pp. 24–24.

Pitman, Jim (1996). "Random discrete distributions invariant under size-biased permutation". In: *Advances in Applied Probability* 28.2, pp. 525–539.

Prasser, Fabian and Florian Kohlmayer (2015). "Putting statistical disclosure control into practice: The ARX data anonymization tool". In: *Medical Data Privacy Handbook*. Springer, pp. 111–148.

Prasser, Fabian, Florian Kohlmayer, and Klaus A Kuhn (2016). "The importance of context: Risk-based de-identification of biomedical data". In: *Methods of information in medicine* 55.04, pp. 347–355.

Prasser, Fabian et al. (2014). "Arx-a comprehensive tool for anonymizing biomedical data". In: *AMIA Annual Symposium Proceedings*. Vol. 2014. American Medical Informatics Association, p. 984.

Prasser, Fabian et al. (2016). "Lightning: Utility-Driven Anonymization of High-Dimensional Data." In: *Transactions on Data Privacy* 9.2, pp. 161–185.

Prasser, Fabian et al. (2017a). "A scalable and pragmatic method for the safe sharing of high-quality health data". In: *IEEE journal of biomedical and health informatics* 22.2, pp. 611–622.

Prasser, Fabian et al. (2017b). "A tool for optimizing de-identified health data for use in statistical classification". In: *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, pp. 169–174.

Prasser, Fabian et al. (2019). "Privacy-enhancing ETL-processes for biomedical data". In: *International journal of medical informatics* 126, pp. 72–81.

Ross, Joseph S, Richard Lehman, and Cary P Gross (2012). "The importance of clinical trial data sharing: toward more open science". In: *Circulation: Cardiovascular Quality and Outcomes* 5.2, pp. 238–240.

Rubner, Yossi, Carlo Tomasi, and Leonidas J Guibas (2000). "The earth mover's distance as a metric for image retrieval". In: *International journal of computer vision* 40.2, pp. 99–121.

Sagiroglu, Seref and Duygu Sinanc (2013). "Big data: A review". In: *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, pp. 42–47.

Schwartz, Paul M (1994). "European data protection law and restrictions on international data flows". In: *Iowa L. Rev.* 80, p. 471.

Sirur, Sean, Jason RC Nurse, and Helena Webb (2018). "Are We There Yet?: Understanding the Challenges Faced in Complying with the General Data Protection Regulation (GDPR)". In: *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*. ACM, pp. 88–95.

Sweeney, Latanya (2000). "Simple demographics often identify people uniquely". In: *Health (San Francisco)* 671, pp. 1–34.

— (2002). "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05, pp. 557–570.

Tankard, Colin (2016). "What the GDPR means for businesses". In: *Network Security* 2016.6, pp. 5–8.

Tenopir, Carol et al. (2011). "Data sharing by scientists: practices and perceptions". In: *PloS one* 6.6, e21101.

Thielman, Sam (2016). "Yahoo hack: 1bn accounts compromised by biggest data breach in history". In: *The Guardian* 15, p. 2016.

Voigt, Paul and Axel Von dem Bussche (2017). "The eu general data protection regulation (gdpr)". In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

Wjst, Matthias (2010). "Caught you: threats to confidentiality due to the public release of large-scale genetic data sets". In: *BMC medical ethics* 11.1, p. 21.

Wong, Raymond Chi-Wing et al. (2006). "(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 754–759.

Zayatz, Laura Voshell (1991). "Estimation of the percent of unique population elements on a microdata file using the sample". In: *Statistical Research Division Report Number: Census/SRD/RR-91/08*. Citeseer.

Zhou, Bin and Jian Pei (2011). "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks". In: *Knowledge and Information Systems* 28.1, pp. 47–77.