INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

ÉCOLE DOCTORALE
ÉNERGIE, MATÉRIAUX, SCIENCES DE LA TERRE ET DE L'UNIVERS
Ou MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE
ET INGÉNIERIE DES SYSTÈMES

[Laboratoire d'Informatique Fondamental d'Orléans]

THÈSE présentée par :

Axel MICHEL

soutenue le : **[08 Avril 2019]**

pour obtenir le grade de : **Docteur de l'INSA Centre Val de Loire**
Discipline/ Spécialité : Informatique

Personalizing Privacy Constraints in Generalization-based Anonymization Models

---

THÈSE dirigée par :
[Benjamin NGUYEN] [Professeur, INSA Centre-Val de Loire]

THÈSE co-dirigée par :
[Philippe PUCHERAL] [Professeur, Université de Versailles Saint-Quentin-en-Yvelines]

RAPPORTEURS :
[Maryline LAURENT] [Professeur, Telecom SudParis]
[Marc-Olivier KILLIJIAN] [Directeur de Recherches, CNRS-LAAS]

JURY :
[Josep DOMINGO-FERRER] [Professeur, Universitat Rovira i Virgili]
[Thi-Bich-Hanh DAO] [Docteur, Université d'Orléans]
[Claude CASTELLUCCIA] [Directeur de Recherches, INRIA Grenoble]

LABORATOIRE D'INFORMATIQUE FONDAMENTALE D'ORLEANS

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

Résumé

Les bénéfices engendrés par les études statistiques sur les données personnelles des individus sont nombreux, que ce soit dans le médical, l'énergie ou la gestion du trafic urbain pour n'en citer que quelques-uns. Les initiatives publiques de *smart-disclosure* et d'ouverture des données rendent ces études statistiques indispensables pour les institutions et industries tout autour du globe. Cependant, ces calculs peuvent exposer les données personnelles des individus, portant ainsi atteinte à leur vie privée. Les individus sont alors de plus en plus réticents à participer à des études statistiques malgré les protections garanties par les instituts. Pour retrouver la confiance des individus, il devient nécessaire de proposer des solutions de *user empowerment*, c'est-à-dire permettre à chaque utilisateur de contrôler les paramètres de protection des données personnelles les concernant qui sont utilisées pour des calculs.

Ce manuscrit développe donc un nouveau concept d'anonymisation personnalisé, basé sur la généralisation de données et sur le *user empowerment*.

En premier lieu, ce manuscrit propose une nouvelle approche mettant en avant la personnalisation des protections de la vie privée par les individus, lors de calculs d'agrégation dans une base de données. De cette façon les individus peuvent fournir des données de précision variable, en fonction de leur perception du risque. De plus, nous utilisons une architecture décentralisée basée sur du matériel sécurisé assurant ainsi les garanties individuelles de respect de la vie privée et de confidentialité des calculs tout au long des opérations d'agrégation.

En deuxième lieu, ce manuscrit étudie la personnalisations des garanties d'anonymat lors de la publication de jeux de données anonymisés. Nous proposons l'adaptation d'heuristiques existantes ainsi qu'une nouvelle approche basée sur la programmation par contraintes. Des expérimentations ont été menées pour étudier l'impact de la personnalisation des contraintes d'anonymat sur la qualité des données. Ces contraintes ont été construites et simulées de façon réaliste en se basant sur des résultats d'études sociologiques.

Mots clés : Protection de la vie privée et des données, Anonymat, Big Data, Matériel sécurisé, Programmation par contraintes.

Abstract

The benefit of performing Big data computations over individual's microdata is manifold, in the medical, energy or transportation fields to cite only a few, and this interest is growing with the emergence of smart-disclosure initiatives around the world.However, these computations often expose microdata to privacy leakages, explaining the reluctance of individuals to participate in studies despite the privacy guarantees promised by statistical institutes. To regain indivuals'trust, it becomes essential to propose user empowerment solutions, that is to say allowing individuals to control the privacy parameter used to mke computations over their microdata. This work proposes a novel concept of personalized anonymisation based on data generalization and user empowerment. Firstly, this manuscript proposes a novel approach to push personalized privacy guarantees in the processing of database queries so that individuals can disclose different amounts of information (i.e. data at different levels of accuracy) depending on their own perception of the risk. Moreover, we propose a decentralized computing infrastructure based on secure hardware enforcing these personalized privacy guarantees all along the query execution process. Secondly, this manuscript studies the personalization of anonymity guarantees when publishing data. We propose the adapation of existing heuristics and a new approach based on constraint programming. Experiments have been done to show the impact of such personalization on the data quality. Individuals'privacy constraints have been built and realistically using social statistic studies.

Keywords: Privacy-preserving, Data privacy and security, Anonymity, Big Data, Secure hardware, Constraint Programming.

Acknowledgement

Firstly, I would like to thank my thesis director Benjamin Nguyen and my co-director Philippe Pucheral. It was a real pleasure to work with them. The many discussions we had at our meetings and advices they gave me resulted in the development of this thesis.

I, also, thank all jury members for their time. I acknowledge the rapporteurs, Maryline Laurent and Marc-Olivier Killijian for their constructive feedback. I also acknowledge Josep Domingo-Ferrer, Thi-Bich-Hanh Dao and Claude Castelluccia to have attended to my defense and for their interesting questions and their feedback on the manuscript.

My gratitude goes to the SDS team of the LIFO for the numerous discussions at coffee breaks. More specifically, I thank Cédric Eichler for his constant help in my papers redaction, my thesis completion and many other things.

I also appreciated the help of Christel Vrain and Thi-Bich-Hanh Dao for their help to understand the concept of constraint programming and their help to the creation of the $k-$anonymity model for constraint programming.

I thank the PETRUS team for their help on the understanding of the Trusted Cells architecture and the embedded DBMS *PlugDB*.

I thank my family for their support during the thesis.

Contents

Résumé **ii**

Abstract **iii**

Acknowledgement **iv**

Contents **v**

List of Figures **ix**

List of Tables **xiii**

1 Introduction (Version Française) **1**
1.1 Motivations 1
1.2 Plan général du manuscrit 6
Contributions 8

2 Introduction (English Version) **9**
2.1 Motivations 9
2.2 Outline of the manuscript 13
Contributions 14

3 State of the art **17**
3.1 Individual's Privacy Concern 18
3.2 Privacy-Preserving Data Publishing 20
3.2.1 Related work on partition-based privacy-preserving methods 21
3.2.2 Related work on $k-$anonymity algorithms 26
3.2.3 Related work on *differential privacy* 31

3.2.4 Related works on personalisation of privacy-preserving methods . . . 37
3.3 Privacy-Preserving in Interactive Query Systems . . . 39
3.4 Data Quality and Information Loss Metrics . . . 43
3.4.1 Metrics in Clustering Techniques . . . 43
3.4.2 Quality Metrics for Anonymisation Techniques . . . 45
3.4.3 Disclosure risk metrics . . . 50
3.5 Conclusion . . . 53

4 Personalised $k-$Anonymity Through SQL 57
4.1 Introduction . . . 58
4.2 Background Informations . . . 59
4.2.1 Reference computing architecture . . . 59
4.2.2 Reference query processing protocol . . . 61
4.2.3 Problem Statement . . . 63
4.3 Personalised Anonymity Guarantees in SQL . . . 63
4.3.1 Modeling anonymisation using SQL . . . 63
4.3.2 The ^{k_i}SQL/AA protocol . . . 65
4.3.3 Differences in importance of attributes . . . 68
4.4 Semantic Issues Linked to K_i-Anonymisation . . . 69
4.5 Algorithmic Issues Linked to K_i-Anonymisation . . . 73
4.5.1 Collection phase . . . 74
4.5.2 Aggregation phase . . . 74
4.5.3 Filtering Phase . . . 78
4.6 Experimental Evaluation . . . 79
4.6.1 Experiments Platform . . . 79
4.6.2 Performance measurements . . . 80
4.7 Conclusion . . . 84

5 Formalisation and Experimentation of Personalised $k-$Anonymity 85
5.1 Introduction . . . 86
5.2 Problem Statement and Scenarios . . . 87
5.2.1 Problem Statement . . . 88
5.2.2 Scenarios . . . 88
5.3 The k_i-anonymity model and linked concepts . . . 89
5.3.1 k_i-anonymity definition . . . 89
5.3.2 Metrics for k_i-anonymity heuristics . . . 90
5.3.3 Personnalized privacy datasets . . . 91
5.4 Personalised $K-$Anonymity Heuristics . . . 93

5.4.1 Mondrian adaptation 93
5.4.2 MDAV and Greedy $k-$member clustering adaptations . 94
5.4.3 $K-$member greedy improvement 95
5.5 Experimenting k_i-anonymity 96
5.5.1 Heuristics comparison with their original 97
5.5.2 Impact of frequencies high/low constraint groups . . . 100
5.5.3 Impact of the scale between high and low constraints . 106
5.5.4 Impact of the conservatives deletion 108
5.5.5 Impact of the correlation between data and privacy concern 109
5.5.6 Recapitulation and conclusion 116
5.6 Synthesis 116

6 Optimal k_i-anonymity with constrained clustering 117
6.1 Introduction 118
6.2 Background and scenarios 119
6.2.1 Clustering 119
6.2.2 Constrained clustering 121
6.2.3 Scenarios 125
6.3 Models 127
6.3.1 $k-$Anonymity model 127
6.3.2 k_i-Anonymity model 130
6.4 Propagations 131
6.4.1 Number of clusters propagator 131
6.4.2 Membership propagator 133
6.4.3 Diameter-based information loss propagator 135
6.5 Branching strategies 137
6.5.1 Clusters creation 138
6.5.2 Filling clusters 140
6.5.3 Switching strategies 141
6.6 Experimental evaluation and limitations 142
6.6.1 Experiment settings 142
6.6.2 Impact of optimized criteria over information loss . . . 143
6.6.3 Limitations discussion 146
6.7 Conclusion 148

7 Conclusion (Version Française) 151
7.1 Contributions 151
7.2 Limitations 155
7.3 Perspectives 158

8 Conclusion (English Version) 161
8.1 Contributions 161
8.2 Limitations 164
8.3 Perspectives 167

Personal publications 171

Figures of chapter 5 173

Bibliography 185

List of Figures

3.1 Privacy concern by age groups 20
3.2 Comparison of pruned lattices merge vs complete generalization lattice 28
3.3 Difference in the creation of equivalence class between MDAV and Greedy $k-$member clustering heuristics 30
3.4 Representation of the differentially private mechanism based on the Laplacian noise (*i.e.* $Lap(|\mathcal{D}|, \epsilon = 1)$) 32
3.5 Partitions of the database by granularity of time (from [27]) . 40
3.6 Minimizing the diameter 44
3.7 Maximizing the space between clusters 44
3.8 Differences between SBIL and DBIL metrics 47
3.9 The EBIL metric weakness: the proximity of different attribute values 48
3.10 Workclass taxonomy tree 50

4.1 Trusted Cells reference architecture [4]. 60
4.2 *SQL/AA* protocol [86, 85] 62
4.3 Regular SQL query form 64
4.4 Example of collection phase with anonymity constraints 65
4.5 ^{k_i}SQL/AA query example 68
4.6 Post-aggregation phase example 69
4.7 Aggregation phase with four partitions. 75
4.8 Merging sorted partitions without NAND access 76
4.9 TDS characteristics [51]. 79
4.10 Generalization tree of `workclass` attribute. 80
4.11 Performance measurements of SQL/AA and state of the art [85]. 81
4.12 Collection Phase Execution Time Breakdown. 82
4.13 Filtering Phase Execution Time Breakdown. 82

4.14 Aggregation Phase Execution Time Breakdown. 83
4.15 TDS Availability Required By The Aggregation Phase. 83

5.1 Correlation between data and privacy concern over a fictitious dataset. 93
5.2 Example of partitionning by Mondrian heuristic 94
5.3 Information loss of heuristics with the personal profile distribution and $k_l = 3$, $k_m = 5$ and $k_c = 7$ 98
5.4 Information loss of heuristics with the personal profile distribution and $k_l = 5$, $k_m = 7$ and $k_c = 10$ 99
5.5 Comparison of k_i-anonymity and k_m-anonymity by varying f_l and f_c with $k_l = 3$, $k_m = 5$ and $k_c = 7$ (scenario 1) 101
5.6 Comparison of k_i-anonymity and k_m-anonymity by varying f_l and f_c with $k_l = 5$, $k_m = 7$ and $k_c = 10$ (scenario 1) 102
5.7 Comparison of k_i-anonymity and k_c-anonymity by varying f_l and f_c ($f_m = 100-(f_l+f_c)$) with $k_l = 3$, $k_m = 5$ and $k_c = 7$ (scenario 2) 104
5.8 Comparison of k_i-anonymity and k_c-anonymity by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 5$, $k_m = 7$ and $k_c = 10$ (scenario 2) 105
5.9 Comparison of k_i-anonymity and k_c-anonymity by varying the scale and k_m on the personal profile distribution (scenario 2) 107
5.10 k_i-anonymity vs k_m-anonymity after removing conservatives, using Greedy $k-$Member (scenario 3) 108
5.11 Privacy concern distribution over age and education 110
5.12 Comparison of k_i-anonymity withoout correlation and k_i-anonymity with correletaion by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 3$, $k_m = 5$ and $k_c = 7$ 111
5.13 Comparison of k_i-anonymity without correlation and k_i-anonymity with correlation by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 5$, $k_m = 7$ and $k_c = 10$ 112
5.14 Box plot of the k_c-anonymity and k_i-anonymity with and without correlated constraints 114

6.1 Creating the most clusters may generate more information loss 129
6.2 Split criterion time line 143
6.3 Diameter criterion time line 144
6.4 WCSD criterion time line 145
6.5 DBIL criterion time line 146

6.6 DBIL criterion 146

List of Tables

3.1 Privacy concern from [2] 18
3.2 Fictitious pseudonymised medical table 21
3.3 Fictitious 3−anonymous medical data table 23
3.4 Minimality attack example [93] 25
3.5 Value of CDR metric on table 3.3 using different set of attributes 51

4.1 Filtering phase 66

5.1 Example of fictitious medical dataset k_i-anonymous. 90
5.2 Privacy concern in function of data category [2]. 92
5.3 Comparison of k_c-anonymity and k_i-anonymity (top left corner without correlation and bottom right with) 115

6.1 Difference between T-B-H Dao et al. model [17], $k-$anonymity model and k_i-anonymity model 127

Chapter 1

Introduction (Version Française)

1.1 Motivations

Les données font aujourd'hui partie intégrante de notre vie. Cette omniprésence des données a donné naissance à une nouvelle science pour les gérer, la *science des données*. Celle-ci est centrale à la méthodologie de nombreux domaines, allant du médical à la sociologie. Elle permet, entre autre, d'avoir une meilleure compréhension des données, ou même la création de nouvelles connaissances. Ces deux bénéfices peuvent être obtenus à l'aide d'études statistiques et de techniques de visualisation des données. En médecine, par exemple, il n'est pas rare d'utiliser des études statistiques pour mettre en évidence des corrélations, permettant de découvrir des causes de maladies et ainsi, mieux les traiter. Avec l'avènement du *Web* et de l'internet des objets (IoT), de nouvelles techniques telles que la fouille de données (ou *data mining*) et l'apprentissage automatique (ou *machine learning*) ont été élaborées pour traiter les jeux de données massifs appelés *Big Data*.

Parallèlement, la *divulgation intelligente* (ou *Smart-disclosure*) consiste à ce que des entreprises et administrations fournissent aux individus les données les concernant, dans un format ouvert. Cela leur permet de les aider dans les décisions qu'ils prennent et, par exemple, réduire leur consommation énergétique. Ce concept est largement encouragé, voire imposé, dans des règlements tels que le RGPD [75] et mis en place par divers consortiums

industriels ou institutionnels. Citons par exemple :

- le *blue button*[1] aux USA, permet aux patients de télécharger leur dossier médical. D'autres industries s'en sont inspirées créant le *green button* pour la consommation énergétique et le *red button* pour les données scolaires.

- le Midata[2], au Royaume-Uni, facilite l'échange des données avec des organismes, par les individus, en échange de connaissances.

- MesInfos[3], en France, accorde la possibilité aux individus de récupérer leurs données auprès des organismes les ayant collectées et leur donne accès à de nouveaux services. Par exemple, dans le cadre de données bancaires, un individu pourrait obtenir des relevés bancaires *intelligents* indiquant sur chaque dépense, les factures, les différents achats faits, ...

D'autre part, l'ouverture des données (*Open Data*) vise à rendre les données utilisables et consultables par tous en les publiant dans un format ouvert. L'ouverture des données est mise en avant par des consortiums tels que le *World Wide Web Consortium*[4] (W3C) et des fondations telles que l'*Open Knowledge International*[5]. La divulgation intelligente et l'ouverture des données promettent un déluge de données amplifiant par la même occasion le *Big Data*. À l'aide de ces deux concepts, les individus se rendent compte de la quantité de données collectées par les divers organismes et des problèmes de vie privée que cela engendre.

À une époque où nous enregistrons tout, la protection de la vie privée est devenu une problématique majeure, comme le démontrent les nombreuses et récentes fuites massives de données personnelles qui ont scandalisé le public. Avec les évolutions technologiques et théoriques dans des domaines tels que l'apprentissage automatique, il est possible de faire de plus en plus de déductions à partir d'un ensemble de données. Ces nouvelles déductions rendent les fuites de données de plus en plus impactantes, fragilisant tout autant la vie privée des individus concernés. Les techniques d'anonymisation se sont répandues pour protéger l'identité des individus concernés. Ces techniques

[1] https://www.healthit.gov/patients-families/your-health-data
[2] https://www.gov.uk/government/news/
[3] http://mesinfos.fing.org/
[4] https://www.w3.org/Consortium/
[5] https://okfn.org/

se basent sur la modification des données pour cacher les individus et ont pour effet, une perte de précision des données appelée *perte d'information*. En général, plus les données personnelles sont anonymisées, plus la perte d'information est importante. Lorsqu'une anonymisation est trop importante, les données anonymes en deviennent inutilisables et, au contraire, une faible anonymisation ne permet pas de protéger les individus. L'objectif des méthodes d'anonymisation est alors de trouver le bon compromis entre perte d'information et anonymat.

Des attaques de désanonymisation ont quand même lieu lorsque des méthodes d'anonymisation sont mal utilisées. Par exemple, *American Online* (AOL) a publié en Août 2006, les historiques de recherches de leurs clients après y avoir appliqué des méthodes d'anonymisation. AOL avait publié les historiques avec un identifiant numérique unique pour chaque utilisateur, méthode couramment appelée *pseudonymisation*. Les recherches d'un même individu étaient liées à un même identifiant et certaines de ces recherches pouvaient elles-mêmes contenir des informations personnelles. Cela a permis de re-identifier bon nombre d'utilisateurs. Thelma Arnold[6], par exemple, a pu être re-identifiée à cause de son amour pour ses chiens, servant maintenant d'exemple emblématique aux problèmes d'anonymisation des historiques de recherche.

La communauté scientifique s'est largement mobilisée autour de la définition de modèles et d'algorithmes d'anonymisation dans l'objectif de rendre la tâche de réidentification plus difficile, tout en minimisant l'impact sur la précision des données. Cependant, le problème de la personnalisation de cette protection est un sujet jusqu'ici peu traité. Il fait pourtant écho au principe de *user empowerment* récemment promulgué dans différentes législations telles que le RGPD. Actuellement, bon nombre de sites proposent une personnalisation des paramètres de confidentialité. En effet, des études montrent que différents individus n'attendent pas d'être protégés de la même façon. Certains admettent pouvoir être peu protégés afin d'améliorer les études statistiques faites et, parallèlement, les services proposés. D'autres estiment qu'une faible protection réduirait trop leurs libertés. Les avis diffèrent donc d'un individu à l'autre et aussi en fonction des données à protéger.

Les gouvernements et le publique portent de plus en plus d'intérêt envers la protection de la vie privée. Un exemple récent prouvant l'intérêt du publique est le scandale lié à *Cambridge Analytica* et *Facebook*. Cambridge

[6]NewYork Times: A Face Is Exposed for AOL Searcher No. 4417749

Analytica avait utilisé une application sur Facebook visant à récolter les données des utilisateurs à des fins de recherche. Malgré que le consentement des utilisateurs était demandé, la façon dont est formé Facebook a permis à Cambridge Analytica de, non seulement, collecter les données des consentants, mais aussi celles de leur cercle social. Des organisations politiques les avaient utilisées dans le but d'influencer l'opinion publique que ce soit pour les élections présidentielles des USA[7], pour le Brexit[8] ou encore, pour les élections présidentielles du Mexique[9]. M. Zuckerberg a dû s'excuser et passer une audition devant le Sénat américain pour répondre des nombreuses atteintes à la vie privée dont Facebook est l'auteur.

D'autre part, l'Union Européenne pousse la protection de la vie privée en promulguant différentes lois limitant les droits des organismes sur les données des utilisateurs. Les institutions privilégiant la protection de la vie privée des individus y voient une manière de développer de nouvelles perspectives technologiques, sociales et économiques. Par exemple, avec le droit à la portabilité, le RGPD [75] propose de faire de la protection de la vie privée un argument de vente de services. Le droit à la portabilité des données permet aux utilisateurs de transférer leurs données personnelles d'un responsable de traitement à un autre. Un responsable de traitement est une personne ou un organisme qui décide de quelle façon sont traitées les données. Le droit à la portabilité donnant la possibilité aux individus de changer de responsable de traitement, un responsable de traitement peut proposer divers services pour obtenir plus de données. Par exemple, il peut proposer de meilleures protections de la vie privée aux individus. Le responsable de traitement peut ensuite effectuer des études sur les données anonymisées ou en obtenant l'accord des individus afin de revendre de la connaissance qu'elles engrangent. L'obtention des données par un responsable de traitement se faisant sur la confiance que les individus lui accordent, proposer des solutions de *user empowerment* est important pour gagner cette confiance.

Actuellement, la façon dont les traitements sont effectués est loin d'être satisfaisante. Par exemple, lorsque l'on souhaite obtenir le salaire moyen par région, un certain nombre d'étapes vont avoir lieu. Un institut de statistique

[7]TheGuardian: Ted Cruz using firm that harvested data on millions of unwitting Facebook users

[8]TheGuardian: Revealed: the ties that bound Vote Leave's data firm to controversial Cambridge Analytica

[9]NewYork Times: Mexico's Hardball Politics Get Even Harder as PRI Fights to Hold On to Power

(*e.g.* l'INSEE), considéré comme un tiers de confiance, collecte tout d'abord les données. Pour cela, il diffuse une requête permettant aux participants de fournir leurs données en indiquant les garanties d'anonymat qui seront satisfaites. Dans un deuxième temps, les utilisateurs vont soit fournir ces données en consentant à répondre à la requête, soit ne rien fournir en cas de non consentement dû à de trop faible garanties d'anonymat. Enfin, l'institut agrège les données et les anonymise afin de pouvoir publier son étude statistique. Cette approche présente deux inconvénients importants :

- Les garanties d'anonymat sont proposées par l'institut et les individus sont anonymisés uniformément. Deux cas peuvent apparaître lors du choix de ce critère uniforme. Dans le premier cas, l'institut peut fournir des garanties d'anonymat trop faibles pour une partie de la population qui ne souhaitera pas participer à l'étude menée par l'institut. Ce manque de participants induit une réduction de la précision de l'étude et les connaissances apportées par cette étude seront moindres. Dans le deuxième cas, l'institut fournira de fortes garanties d'anonymat permettant d'obtenir la totalité des participations. Cependant, la perte d'information sera élevée. Les participants acceptant de faibles garanties seront de fait, anonymisés plus que nécessaire.

- Les instituts ou organismes opérant ce genre de collectes et de calculs sont souvent considérés de *confiance*. Les calculs et les données sont centralisés rendant le bénéfice d'une attaque, le coût d'une erreur, l'impact d'une faille plus important. Le cas du piratage *Yahoo*[10] en est un parfait exemple. Alors que Yahoo est une compagnie ayant les capacités pour sécuriser les données, un groupe de pirates avait réussi en 2014 à récupérer des informations personnelles de plus d'un milliard de comptes. Leur attaque était peut être coûteuse à effectuer mais le bénéfice en était tout aussi élevé. En général, les données personnelles collectées par un organisme sont stockées sur un serveur centralisé et, malheureusement, toute attaque sur une solution centralisée génère un bénéfice tel qu'elle est toujours rentable, augmentant ainsi sa probabilité. À défaut de pouvoir empêcher tout piratage, utopie surréaliste arguée par bon nombre de compagnies, il devient nécessaire de réduire le ratio bénéfice/coût de ces attaques pour dissuader les pirates.

[10] https://www.bbc.co.uk/news/world-us-canada-37447016

Le **premier objectif** de ce manuscrit est de proposer un nouveau concept d'anonymisation permettant aux individus de spécifier des contraintes d'anonymat personnalisées. Des solutions algorithmiques doivent également être proposées pour garantir l'applicabilité de ces contraintes, palliant ainsi le premier inconvénient présenté ci-dessus. Ce concept représente une solution de *user empowerment* pouvant être utilisé par les responsables de traitement et les instituts de statistiques.

Le **deuxième objectif** de ce manuscrit est de proposer une architecture décentralisée pour éparpiller les données et réduire l'impact d'une potentielle fuite de données. Ce système devra être capable d'effectuer des calculs sur les données de façon sécurisée. Les utilisateurs auront plus de contrôle sur leurs données. Il sera possible de les contacter pour participer à une étude statistique à laquelle il décideront d'y participer ou non. Dans le cas où un individu participe, les données ne seront pas fournies à l'entité demandeuse mais l'analyse sera effectuée de manière sécurisée et distribuée au sein de cette architecture.

1.2 Plan général du manuscrit

Ce manuscrit se décompose en cinq parties. Le chapitre 3 présente les principales approches de l'état de l'art relatives aux sujets étudiés par ce manuscrit. Premièrement, ce chapitre présente un ensemble d'études faites montrant la différence d'opinion que les individus ont sur la protection de la vie privée. Puis, les méthodes d'anonymisation des données dans l'objectif de les publier tout en respectant le vie privée des individus, sont présentées. Le $k-$anonymat présente des caractéristiques nécessaires au *user empowerment*. Tout d'abord, le $k-$anonymat permet l'anonymisation de données sans restreindre les opérations applicables sur ces données. Ensuite, la compréhension de ce concept d'anonymisation est à la portée de n'importe quel individu, caractéristique importante du *user empowerment*. C'est pourquoi, ce manuscrit se focalise sur le $k-$anonymat et les différentes heuristiques permettant de l'appliquer, sont présentées par le chapitre 3. Il décrit aussi les systèmes proposant de faire des calculs sur les données personnelles sans pour autant affecter la vie privée des individus concernés par ces données. Finalement, ce chapitre expose les différentes métriques permettant de mesurer la perte d'information induite par l'anonymisation d'un jeu de données et, ce faisant, la qualité des données.

Le chapitre 4 introduit une première contribution. En premier lieu, ce chapitre décrit un moyen d'incorporer des garanties d'anonymat personnalisées permettant aux utilisateurs de participer à des études statistiques en fonction de leur opinion sur la protection qu'ils veulent et sur les garanties de protection assuré par une requête. Ce chapitre se base sur l'utilisation d'une architecture permettant à un organisme d'effectuer des calculs sur les données personnelles sans que cet organisme puisse voir des résultats intermédiaires et tout en assurant que le résultat de ces calculs soit en accord avec les contraintes d'anonymat fournies par les utilisateurs. Finalement, ce chapitre effectue des expérimentations sur un jeu de données réelles et de grande taille montrant l'efficacité de l'approche présentée dans ce chapitre.

Le chapitre 5 expose une autre contribution basée sur une pratique commune, la publication de jeux de données anonymisés. Tout d'abord, ce chapitre formalise le concept de $k-$anonymat personnalisé. Ensuite, il propose une adaptation des algorithmes existants pour produire un $k-$anonymat personnalisé rendant l'application de ce nouveau concept plus aisée. Il propose aussi une discussion sur la différence d'opinion que les individus ont sur les contraintes d'anonymat qu'ils estiment être raisonnables. Cette opinion étant corrélée au vécu des individus, il peut être difficile d'estimer l'intérêt du $k-$anonymat personnalisé dans certains scénarios. Ce chapitre permet d'identifier les cas d'usages pour lesquels l'intérêt de la méthode est particulièrement important. Enfin, il propose une série d'expériences révélant les avantages, l'impact de la corrélation et les limites de l'approche proposée dans ce chapitre.

Le chapitre 6 présente une nouvelle approche, basée sur la programmation par contraintes, pour calculer le $k-$anonymat personnalisé optimal. Cette nouvelle approche peut s'adapter facilement aux différents concepts d'anonymisation de données (*e.g.* $k-$anonymat, $\ell-$diversité, ...) induisant de nouvelles perspectives. Le chapitre expérimente cette nouvelle approche sur des jeux de données de faible taille correspondant à un ensemble restreint de scénarios mais néanmoins réels. Enfin il discute des limites de cette nouvelle méthode indiquant les principales améliorations à effectuer pour augmenter sa capacité en terme de taille de jeux de données.

Enfin, le chapitre 7 conclut. Il résume les différentes contributions faites dans ce manuscrit et expose les perspectives offertes par ces mêmes contributions.

Contributions

Ce manuscrit présente un certain nombre de contributions. Ces contributions sont résumées et listées ci-dessous :

- la présentation d'un nouveau concept d'anonymat permettant aux individus de spécifier le degré d'anonymat qu'ils souhaitent. Ce nouveau concept a été expérimenté sur des scénarios réalistes montrant une réduction de la perte d'information dans la plupart des scénarios. Des heuristiques existantes ont été adaptées à ce nouveau concept et utilisées pour ces expérimentations. De plus, ces expérimentations montrent aussi que la corrélation entre les contraintes d'anonymat des individus et leurs caractéristiques (*e.g.* leur âge, ...), améliore la qualité des données.

- l'élaboration d'un algorithme permettant la collecte et le calcul distribué de requêtes avec des garanties d'anonymat personnalisées sur de larges quantités de données. Cet algorithme prend en compte les différentes sémantiques possibles liées à la personnalisation de l'anonymat.

- la création d'une nouvelle méthodes pour anonymiser un jeu de données avec l'aide de la programmation par contraintes. Les contraintes, exprimées sous forme d'un modèle, permettent d'obtenir une solution optimale d'anonymisation personnalisé. Plusieurs critères d'optimisation communs aux techniques de *clustering* ont été utilisés et évalués expérimentalement.

Chapter 2

Introduction (English Version)

2.1 Motivations

Nowadays, data are a main part of our society. This data ubiquity gave birth to a new science, the *data science*. Data science is central to many science fields' methodology, ranging from medicine to sociology. It allows scientists to get a better understanding of data shape and allows new knowledge to be created. This two benefits are obtained with the help of statistic studies and data visualization methods. In the medical field, for example, the use of statistic studies is generally a good way to highlight correlations which show causes of illnesses and how to improve their treatment. With the advent of the *Web* and the *Internet of Things* (IoT), new methods such as data mining, and machine learning have been studied to scale to *Big Data*.

At the same time, *Smart-disclosure* consists to allow users to obtain data in an open format that companies and administrations collect on them. The goal is to help users to make their decisions. For example, they can use a detail of their energy consumption to reduce it. The concept of smart-disclosure is widely pushed forward, or enforced by regulations such as the GDPR [75], and proposed by industry-led or institutional consortia. We can cite some *Smart-disclosure* initiatives :

- the *blue button*[1], in USA, allows patients to get their medical records. The *blue button* has inspired others smart-disclosure concept such as the *green button* for the energy consumption and the *red button*, for education data.

[1] https://www.healthit.gov/patients-families/your-health-data

- the *Midata*[2], in UK, facilitates the share of data by users with companies in exchange of knowledge.

- MesInfos[3], in France, lets users to get data from companies collecting those and gives them access to new services. For example, they can use bank data to obtain *smart* bank statement, linking to each expenses the invoice, the list of purchases, ...

On the other hand, the *Open Data* aims to leave data usable and searchable for all by publishing them in an open format. The open data is pushed forward by consortia such as the *World Wide Web Consortium*[4] (W3C) and foundation such as the *Open Knowledge International*[5]. The smart-disclosure and the open data promise a deluge of data increasing even more, benefits from *Big Data*. Both concepts help users to realize the amount of data collected by companies and various organizations, and the resulting privacy issues.

At a time every data are collected, privacy-preserving has become a major issue as demonstrated by the many recent data leaks scandalizing the public. Due to technological and theoretical development such as machine learning, more and more inferences can be made from a dataset. These inferences are making data leaks more impacting, weakening the privacy of concerned individuals. Anonymisation methods have become a common practice to protect individuals' identity. These methods are based on data distortions and lead to a loss of data accuracy called *information loss*. Generally, the more data are anonymised, the more information loss is generated. An excessive anonymisation leads to anonymised data without utility, and contrariwise, a low anonymisation does not protect individuals. The goal of anonymisation methods is to propose a trade off between information loss and privacy.

Deanonymisation attacks still occur because of anonymisation misuses. For example, in August 2006, *American Online* (AOL) had published their client search data after using some anonymisation methods. AOL had published search data with a unique identifier for each of their users, a methods called *pseudonymisation*. Searches of the same individual were linked to the same identifier and some of these searches could contain personal information.

[2]https://www.gov.uk/government/news/
[3]http://mesinfos.fing.org/
[4]https://www.w3.org/Consortium/
[5]https://okfn.org/

This allowed people to re-identify many users. Thelma Arnold[6], for example, have been re-identified because of her love for her dogs, now serving as an iconic example of the difficulties of anonymising search data.

The definition of anonymisation models and algorithms has been a main concern for number of scientists in order to make re-identification harder while minimizing the impact of anonymisation on data utility. However, the personalisation of individuals protection is a topic that has not been developed enough. Yet, it echos *user empowerment* concepts that are pushed forward by legislators such as the GDPR. Currently, number of websites offer the personalisation of confidentiality parameters. Indeed, some studies show that different individuals do not expect to be protected the same way. Some agree to be less protected to improve statistics and services. In the other hand, others believe that a low protection would lead to the loss of their freedoms. Therefor, opinions differ from one individual to another and also according to the data involved.

Governments and the public show more and more interest about privacy. A recent example about this interest is the scandal of *Cambridge Analytica* and *Facebook*. Cambridge Analytica used an application on Facebook which goal was to collect data for research purpose. Even if users' consent were asked, the way Facebook is built, allowed Cambridge Analytica to collect data from consenting but also and mostly data from users in the social circle of consenting users. Those data were used by political organizations to influence public opinion whether for the US presidential elections[7], for the Brexit[8], or for Mexico's presidential elections[9]. Mr Zuckerberg had to apologize and was heard by the US Senate to respond to the many violations of privacy that Facebook is responsible.

On the other hand, the European Union enforces privacy of individuals by enacting number of laws to limit data misuses by organisms collecting Europeans data. Institutions that ensures privacy preservation, see it as a way to develop new technological, social and economic opportunities. For example, the GDPR [75], introduces the *right to data portability* which can

[6]NewYork Times: A Face Is Exposed for AOL Searcher No. 4417749

[7]TheGuardian: Ted Cruz using firm that harvested data on millions of unwitting Facebook users

[8]TheGuardian: Revealed: the ties that bound Vote Leave's data firm to controversial Cambridge Analytica

[9]NewYork Times: Mexico's Hardball Politics Get Even Harder as PRI Fights to Hold On to Power

be use to make privacy-preserving methods a selling argument. The right to data portability allows users to transfer their data from a data controller to another. The data controller is a person or an organism in charge of choosing the way data are processed. For example, it can ensure better privacy protections to attract more individuals. Then, the data controller can compute statistics on anonymised data and sell knowledge it generates. To get data from individuals, the data controller should increase the trust of individuals. This trust can be increased by proposing *user empowerment* solutions for example.

The way micro-data is anonymised and processed today is far from being satisfactory. Let us consider how a national statistical study is managed, *e.g.* computing the average salary per geographic region. Such a study is usually divided into 3 phases: (1) the statistical institute (assumed to be a *trusted third party*) broadcasts a query to collect raw micro-data along with *anonymity guarantees* (*i.e.*, a privacy parameter like k in the case of $k-$*anonymity* or ϵ in the case of *differential privacy*) to all users ; (2) each user consenting to participate transmits her micro-data to the institute ; (3) the institute computes the aggregate query, while respecting the announced anonymity constraint.

This approach has two important drawbacks:

1. The anonymity guarantee is defined by a querier (*i.e.* the statistical institute), and applied uniformly to all participants. If the querier decides to provide little privacy protection, it is likely that many users will not want to participate in the query leading the study to a loss of accuracy. On the contrary, if the querier decides to provide a high level of privacy protection, many users will be willing to participate, but the quality of the results will drop. Indeed, higher privacy protection is always obtained to the detriment of the quality and thus utility of the sanitized data.

2. The querier is assumed to be trusted. Although this could be a realistic assumption in the case of a national statistics institutes, this means it is impossible to outsource the computation of the query. Moreover, micro-data centralization exacerbates the risk of privacy leakage due to piracy (Yahoo and Apple recent hack attacks are emblematic of the

weakness of cyber defenses[10]), scrutinization and opaque business practices. This erodes individuals trust in central servers, thereby reducing the proportion of citizen consenting to participate in such studies, some of them unfortunately of great societal interest.

The **first goal** of this manuscript is to propose a new anonymity concept which allows individuals to specify anonymity constraints. Algorithms are also presented to guarantee these constraints while anonymising individuals to overcome the first drawback. This new concept is a *user empowerment* solution which can be used by data controller and statistics institutions.

The **second goal** of this manuscript is to propose a decentralize architecture to scatter the data and so, decrease the impact of a data leak. This system would be able to securely compute statistics. In this system user would have control over their own data. They would be able to choose to participate to statistics studies or not. In the case the individual is willing to participate in statistics, the querier would be able to see its data but only aggregation computed securely by the architecture.

2.2 Outline of the manuscript

This manuscript is divided into five parts. The chapter 3 presents the different approaches of the state of the art related to the main subject of this manuscript. Firstly, this chapter shows studies highlighting the difference of individuals' opinions about privacy. Then, the chapter shows different methods of data anonymisation for the purpose of data publishing without privacy leak. The $k-$anonymity, one of these methods, allows data anonymisation without restricting computations we can make over it. It is also a concept understandable by any individual which is an important feature to propose *user empowerment* solution. That is why this manuscript focuses on the $k-$anonymity and the different existing heuristics to apply it are presented by the section 3. Then, the chapter exhibits different systems which allow computations over data such as statistics without lowering the privacy of individuals. Finally, the chapter presents a set of metrics made to compute the information loss induced by data anonymisation and, so, the data quality.

[10]Yahoo 'state' hackers stole data from 500 million users - BBC News. https://www.bbc.co.uk/news/world-us-canada-37447016

The chapter 4 introduces a first contribution. Firstly, it describes a way to add personalised anonymity guarantees allowing users to participate to statistics studies depending on their opinion about the security they want and the guarantees ensured by the query. This chapter is based on an architecture which allow an organisation to make computations over users personal data without access to any intermediate result while ensuring that users privacy constraints are satisfied regarding the result. Finally, this chapter shows experiments made on a real and big dataset exhibiting the efficiency of the approach.

The chapter 5 shows another contribution based on a common approach, Privacy-Preserving Data Publishing (PPDP). Firstly, it defines the personalised $k-$anonymity concept. Then, it presents adaptation of heuristics made for the $k-$anonymity to produce personalised $k-$anonymous datasets. Then, it proposes a discussion about the difference of opinions individuals got about anonymity constraints they consider to be reasonable. Since this opinion is correlated to their lived experiences it may be difficult to estimate the benefits from such an approach. This is why the chapter also identifies use cases in which the personalised $k-$anonymity is particularly profitable. Finally, it proposes a set of experiments revealing the benefits, the impact of data correlations and the limits of the methods presented by this chapter.

The chapter 6 presents a new methods to compute optimal personalised $k-$anonymity using constraint programming. This new method easily fit to number of data anonymisation concepts (*e.g.* $k-$anonymity, $\ell-$diversity, ...) leading to new perspectives. The chapter experiments this new approach on small datasets corresponding to a limited set of scenarios and yet real. Finally, it discusses the limits of this approach and shows the main steps to improve to increase the scalability of the approach.

Finally, the chapter 8 concludes. It summarizes the different contributions presented in this manuscript and exhibits the resulting perspectives.

Contributions

This manuscript presents some contributions. These contributions are summarized and listed below:

- the presentation of a new anonymity model allowing individuals to specify the anonymity level they wish. This new concept has been

tested on realistic scenarios showing an improvement of data quality in most of the scenarios. Existing heuristics have been adapted to this new approach and evaluated through a multitude of experiments. These experiments also show that the correlation between individuals anonymity constraints and their own characteristics (*e.g.* age, weight, ...) is improving the overall data quality.

- the development of an algorithm that can make distributed computations of query with personalized anonymity guarantees on large datasets. This algorithm take into account the different semantics related to personalised anonymity.

- the creation of a new anonymisation methods based on constraint programming. Constraints are expressed as models and make it possible to compute optimal personalised anonymity solution. Some optimisation criteria commonly used by *clustering* methods are used and experimentally evaluated.

Chapter 3

State of the art

This chapter presents background knowledge and related work on the subjects studied by this thesis.

The first part presents statistic and psychological studies showing the variation of individuals' opinions about privacy. It shows how much the personalisation of privacy-preserving methods is important for individuals and offers great user empowerment perspectives.

The second part introduces Privacy Preserving Data Publishing (PPDP) methods. Two main approaches are discussed. The partition-based approach (*i.e.* $k-$anonymity based methods) is presented first, and the *differential privacy* concept second. Since this thesis focuses on partition-based approaches, a state of the art of $k-$anonymity algorithms is presented. Related works on the personalisation of the two approaches are presented at the end of this part.

After that, we talk about privacy-preserving in interactive query systems. We present existing solutions to ensure good privacy preservation while answering interactive queries. Some computing architectures reaching this goal are presented in this part.

Finally, we focus on metrics used to measure data quality and information loss of modified data produced by PPDP methods. We present metrics used from two different domains which share similarities: clustering techniques for data mining and anonymisation methods.

	General privacy concern	Data about offline identity	Data about online identity	Data about personal profile	Data about professional profile	Data about sexual and political identity
High concern	53.7 %	39.6 %	25.2 %	0.9 %	11.9 %	12.1 %
Medium concern	35.5 %	48.3 %	41.2 %	16.8 %	50.8 %	25.8 %
Low concern	10.7 %	12.1 %	33.6 %	82.3 %	37.3 %	62.1 %

Table 3.1: Privacy concern from [2]

3.1 Individual's Privacy Concern

The GDPR [75] tries to push user empowerment initiatives. The right to data portability enacted in this new regulation allows users to obtain their personal data from a data controller and transfer them to another data controller. It gives them the possibility to choose a data controller which matches their expectations. Thus it is expected that a user will gain much more control over his data. It is important to note that users' expectations (*i.e.* privacy concerns) are different from one to another. To understand privacy concerns behaviour, a number of researchers have studied the privacy concerns of individuals concerning different types of data. This section is dedicated to existing studies on privacy concerns.

Privacy concern is a topic that has been studied for a long time. H. Jeff et al. have presented a state of the art [79] of studies presented before 2000. Some of studies from this paper date back to 1972. Since global privacy concern evolves with time, we will give more interest to recent studies. A study by A. Acquisti and J. Grossklags [2] about different kinds of data used shows that even for the same person, privacy concern is something that depends of the usage of personal data. Table 3.1 shows results presented on [2].

Table 3.1 represents the privacy attitude of people depending on the category of personal information. For example, the category *Data about personal profile* shows the privacy concern for profiling information such as age, weight or isolated piece of personal information. On the other side, the *General*

privacy concern category shows the privacy concern that subjects' studied generally expressed. Acquisti et al. could identify four groups of persons differing by their privacy concerns. One group showed a high concern about privacy, and two groups showing a moderate concern have been merged into one group. The last group expressed a low privacy concern. In their study, A. Acquisti and J. Grossklags identified that individuals were more concerned about privacy when identifying information was requested than with profile information such as age or weight. People also showed a higher concern when the pieces of information queried were connected together rather than with isolated pieces of information.

Another important result demonstrated by their paper, is the correlation between data and privacy concern. They asked the importance people express for privacy and then ask for privacy concern information. Obviously, people showing higher importance to privacy showed also higher privacy concern than people finding privacy less important. However they also noted that in the same group, some people saying privacy was important tend to express lower privacy concern. They found that *privacy attitudes appear correlated with income; people with low income tend to be less concerned about privacy.* As income is correlated with different pieces of personal information (*e.g.* age, sex), privacy concern can also be correlated to a set of personal information. This being said, it is important to note that actions of an individual are often contradictory with their opinion about privacy concern.

Another study presented by E. Van den Broeck et al. [13] shows that age was correlated with privacy concern. They identified a variation of privacy concern between three different groups. The group with the highest privacy concern was composed of middle-aged adults (*i.e.* 40–65 years old). Young adults (*i.e.* 25–40 years old) expressed a lower privacy concern than middle-aged adults. The lowest privacy concern was expressed by the emerging adult (*i.e.* 18–25). Figure 3.1 illustrates the results of this study by plotting normal distribution with mean and standard deviation given in their paper [13].

H. K. Tsoi and L. Chen studied the influence of nationality by looking at the differences between Hong Kong people and French people in term of privacy concerns [87] in social network sites. They showed that french and Hong Kong users expressed different degrees of privacy concern. Following this idea, it is easy to suppose there exists a correlation between privacy concern and the nationality. They also measured the trust of users in social network site exhibiting a correlation between this trust and privacy concern. By generalizing this last information, it is easy to suppose that increasing

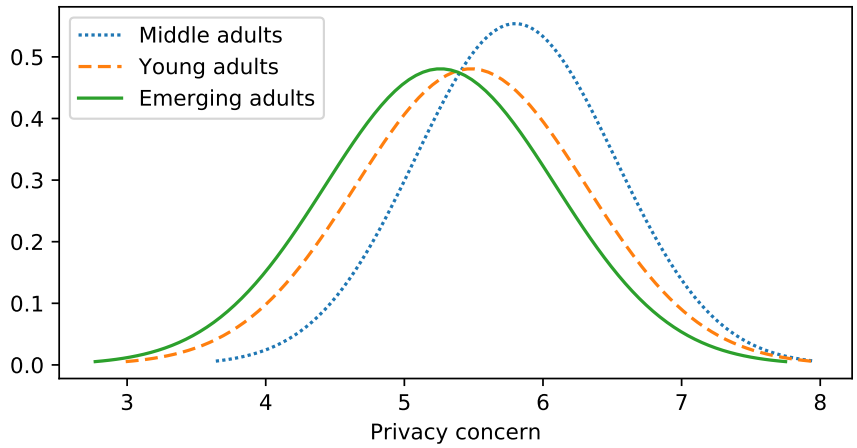


Figure 3.1: Privacy concern by age groups

the trust of users in data controllers would lead to increasing the amount of data users would be willing to share.

Since privacy concern varies highly between people, it is important to give users the possibility to express their opinion and to take into account their difference. Privacy concern is correlated with many different factors, such as quality of life, nationality, income, age and certainly other factors not yet studied (*e.g.* having faced a serious illness, a divorce or losing one's job).

From the viewpoint of the authorities trying to acquire data for statistical studies, it is important to note that, trying to respect the different opinions may facilitate individuals to consent to their participation in statistical studies. This would lead to an increase in the amount of collected data and, in consequence to an increase in accuracy of a statistical study, which can be viewed as one of the motivations of this thesis.

3.2 Privacy-Preserving Data Publishing

Privacy-Preserving Data Publishing (PPDP) has been a hot topic for the last twenty years. The objective of PPDP techniques is to publish data while ensuring the privacy of individuals. Individuals must be protected against an attacker who is going to lead a de-anonymisation attack. The attacker tries to

Identifier	Quasi-identifier			Sensitive
Name	ZIP	Age	Sex	Condition
A	14025	25	F	Cancer
B	14025	32	M	Cancer
C	14020	35	M	Heart Disease
D	14110	38	F	Cancer
E	14100	39	M	Viral Infection
F	14110	44	F	Viral Infection
G	14110	70	M	Heart Disease
H	14020	70	M	Viral Infection
I	14025	50	F	Cancer

Table 3.2: Fictitious pseudonymised medical table

link sensitive information such as medical data or salary to a physical person. To prevent the attacker from inferring sensitive information of individuals, two main approaches have emerged:

- partition-based approaches
- *differential privacy* approaches

3.2.1 Related work on partition-based privacy-preserving methods

Before 2002, pseudonymisation was the most used approach to protect privacy. Pseudonymisation consists in replacing data which directly identifies an individual (*i.e.* first and last name) by a pseudonym. The pseudonym can be reverted to its original value with the help of another piece of information (*e.g.* a secret key in case of encryption). Table 3.2 illustrates a fictitious medical table that has been pseudonymised : The names of people involved have been replaced by pseudonyms.

In 2002, L. Sweeney presented an attack on the pseudonymisation approach [83]. This attack was based on the use of different databases to link pieces of information together. She bought the voter registration list of Cambridge Massachusetts for twenty dollars and got a pseudonymised dataset from the Group Insurance Commission (GIC) which was given for free to researchers. This last dataset was supposed to be anonymous due to

pseudonymisation. By crossing the two datasets, she managed to identify the governor of Massachusetts, William Weld in the GIC dataset and recover his medical information. She needed only three attributes to re-identify the governor: ZIP code, sex and birth date. Those kind of attacks are known as background knowledge attacks due to the fact that the attacker has some knowledge of the value of some attributes of an individual (other than identifiers). To overcome the weakness of pseudonymisation, L. Sweeney proposed a new way to protect people from re-identification : $k-$anonymity [83]. $K-$anonymity is a partition based approach to anonymise a dataset. It consists of generalization and suppression of different attributes values. In PPDP, we can distinguish three kind of data: *identifiers*, *quasi-identifiers* and *sensitive data*.

Definition 1 (Identifier, Quasi-identifier and Sensitive data)**.** We denote by *identifier* data that directly identifies a physical individual (*e.g.* First Name, Last Name, ID Number). We denote by *quasi-identifiers* a set of attributes that can identify *some individuals* when grouped together (*e.g.* ZIP code, birth date, sex). Finally, we denote by *sensitive* information an attribute whose value we do not want to be able to link back to a physical person (*e.g.* salary, medical diagnosis).

$K-$anonymity aims to regroup records with similar quasi-identifiers into groups of size k. Records of a same group are generalized to share the same quasi-identifier. We call it an *equivalence class*.

Definition 2 (Equivalence class)**.** Let $\mathcal{D}$ be a set of personal data records. An equivalence class e is a set of records such as $e \subseteq \mathcal{D}$ and two records are in the same equivalence class iff they have similar quasi-identifiers.

Table 3.3 illustrates the previous fictitious medical dataset as a $k-$anonymous table (*i.e.* with $k = 3$). Identifiers have been shown to simplify the explanation but obviously $k-$anonymity removes them. The goal of creating equivalence classes is that if an attacker has background knowledge about some users in the database, he can find in which group those users are but cannot infer their sensitive information. For example, in table 3.3, if the attacker knows that Bob is a 32 years old male with zipcode 14025, the attacker can only say that Bob has a 66% chance to have a cancer, and a 33% chance to have heart disease.

Identifier	Quasi-identifier			Sensitive
Name	ZIP	Age	Sex	Condition
Alice	1402*	$[25, 35]$	*	Cancer
Bob	1402*	$[25, 35]$	*	Cancer
Charlie	1402*	$[25, 35]$	*	Heart Disease
Diana	141**	$[38, 44]$	*	Cancer
Eric	141**	$[38, 44]$	*	Viral Infection
Fany	141**	$[38, 44]$	*	Viral Infection
Gregor	14***	$[45, 70]$	*	Heart Disease
Henry	14***	$[45, 70]$	*	Viral Infection
Irene	14***	$[45, 70]$	*	Cancer

Table 3.3: Fictitious $3-$anonymous medical data table

One of the weaknesses of $k-$anonymity is that there is no control over the sensitive attributes. If all individuals of an equivalence class have cancer, an attacker identifying that a user is in this equivalence class would know that this user has cancer. This attack is called the homogeneity attack. A. Machanavajjhala et al. have proposed another concept based on the $k-$anonymity approach to prevent homogeneity : $\ell-$diversity [61]. The $\ell-$diversity approach is similar to $k-$anonymity but ensures in addition that all equivalence classes have at least ℓ distinct sensitive values. In our example, table 3.3 is also $2-$diverse because any equivalence class contains at least two distinct sensitive values. A. Machanavajjhala et al. proposes different instantiations of $\ell-$diversity (*e.g.* based on entropy) to fit different problems (*e.g.* in case a sensitive value is much more frequent than others).

The $\ell-$diversity approach also has weaknesses. In our example, let the cancer be 50% of the diseases of our database and 25% for viral infection and heart disease. We know for sure that Bob does not have a viral infection which leads to a leak of information. The ideal protection of a user is to be in an equivalence class containing 50% of cancers, 25% of heart disease and 25% of viral infection because it will give no extra information than knowing a user is in the database. This is the goal of $t-$closeness, an approach proposed by N. Li et al. [58]. The $t-$closeness approach constructs equivalence classes which have similar frequency of sensitive values to the whole database. This concept involves much higher degradation of data due to the fact that many correlations exist between quasi-identifier and sensitive values. For example,

heart disease may be correlated to age [46] and living near the location of a nuclear incident may lead to a population with more cancers. Some disease are also correlated to the sex of an individual (*e.g.* breast cancer). Trying to keep the same frequency of sensitive values in all equivalence classes leads to the loss of such correlation information, which is usually of paramount importance in statistical studies. This is why it is important to find the right trade off between privacy and data quality.

Having restriction on the frequency of sensitive attributes is a way to prevent an attacker from conducting frequency attacks but may also make it possible to use another attack, the *minimality attack*. The minimality attack, presented by R.C. Wong et al. [93], uses the fact that some generalizations are done to satisfy constraints on the sensitive attribute. Let us take the $\ell-$diversity as example. In the table 3.3, if we aim for $3-$diversity, only the third equivalence class (with Gregor, Henry and Irene) satisfies this constraint. The first and second equivalence class need others generalizations to ensure they have 3 distinct sensitive values. An attacker could use this information to re-identify users. To illustrate it with a simpler example, let us use the example proposed in their paper [93]. Table 3.4 shows a dataset which will be generalized to achieve $2-$diversity. In this table, we have two groups of people with the same quasi-identifier and the same sensitive value. To achieve $2-$diversity, the two groups will be merged into one group with the same quasi-identifier. If the attacker knows that there were two people with q_1 quasi-identifier and three with q_2 as quasi-identifier, the attacker can recreate the table 3.4 which does not satisfy the $2-$diversity condition. To override this weakness, they proposed another partition-based concept, $m-$confidentiality. The minimality attack uses background knowledge represented by another anonymised table and links probabilistically sensitive values to records. The $m-$confidentiality approach aims to reduce the accuracy of these probabilities. In fact, it consists in adding randomness to other partition-based approaches (*i.e.* not trying to obtain optimal solutions). To illustrate their approach, they made an algorithm which builds an $\ell-$diverse table. This table can be specialized to produce another $\ell-$diverse table (*i.e.* because it is not optimal).

Many concepts exist and they all aim to protect individuals against different attack models [29]. There are two families of attack models against partition-based privacy preserving methods: *linkage* attacks and *probabilistic* attacks [29]. We can distinguish three levels of linkage attacks: *record* linkage, *attribute* linkage and *table* linkage. They differ by the knowledge the

Quasi-identifier	Sensitive
q_1	Viral Infection
q_1	Viral Infection
q_2	Cancer
q_2	Cancer
q_2	Cancer

Table 3.4: Minimality attack example [93]

attacker has.

Record linkage is the attack that the $k-$anonymity approach tries to prevent. The attacker knows that the record of a physical individual is in the anonymised data table. He also knows the quasi-identifiers of this individual. The record linkage attack consists in using this information to re-identify which record corresponds to the person targeted (without taking sensitive values into account). Since there are k persons with the same quasi-identifiers, the attacker cannot tell which is the targeted individual among the k.

The attribute linkage attack aims to infer the sensitive values of an individual without re-identifying precisely which record corresponds to that person's record. For example, the homogeneity attack does not identify the record but permits to stick a sensitive values to a physical person. The $\ell-$diversity aims to prevent the attribute linkage attack. The two first linkage attacks are based on the fact that the attacker knows that the targeted record is in the anonymised table.

The third model, table linkage, assumes that the attacker does not know if the targeted record is in the anonymised table. The attacker has a knowledge from a public database $\mathcal{P}$. Let $\mathcal{D} \subseteq \mathcal{P}$ be a dataset with $\mathcal{D}^*$ an anonymisation table. The attacker can make an assumption about the fact that a record from $\mathcal{P}$ is present in $\mathcal{D}^*$. To prevent the attacker from completing such a table linkage attack, M. E. Nergiz and C. Clifton proposed a new privacy model, $\delta-$presence which limits the confidence of the assumption made by the attacker. The attacker should not infer a probability out of certain bounds given in equation 3.1.

$$\delta_{min} \leq Pr\left[r \in \mathcal{D} \mid \mathcal{D}^*, \mathcal{P}\right] \leq \delta_{max} \tag{3.1}$$

with $\delta = (\delta_{min}, \delta_{max})$.

The second family of attacks (*i.e. probabilistic* attacks) aims to improve the knowledge about sensitive information of a record. Let us take the example from $\ell-$diversity. The attacker could deduce that Bob has a 66% chance to have cancer and 33% to have heart disease. Assume the attacker knows that there is 50% of cancer in the dataset, the anonymised table would give knowledge which differs from his original belief. That is what $t-$closeness [58] tries to prevent.

3.2.2 Related work on $k-$anonymity algorithms

Since most partition-based models are based on $k-$anonymity, this section presents a set of algorithms to achieve $k-$anonymity. Those algorithms differ in their objective : some try to improve a metric of information loss when others try to decrease processing time required to produce the anonymised dataset.

The first $k-$anonymity algorithm was proposed by L. Sweeney and P. Samarati and was called *MinGen* [77]. It was actually done before $k-$anonymity was formalized. It was a theoretical algorithm. The goal of MinGen was to produce every generalization of the dataset to anonymise and select the generalization which grants $k-$anonymity with the lowest information loss. This algorithm gives, obviously, an optimal solution but is not feasible due to the number of possible generalizations of a dataset. This is due to the fact that *Optimal $k-$anonymity* is an NP-hard problem [72].

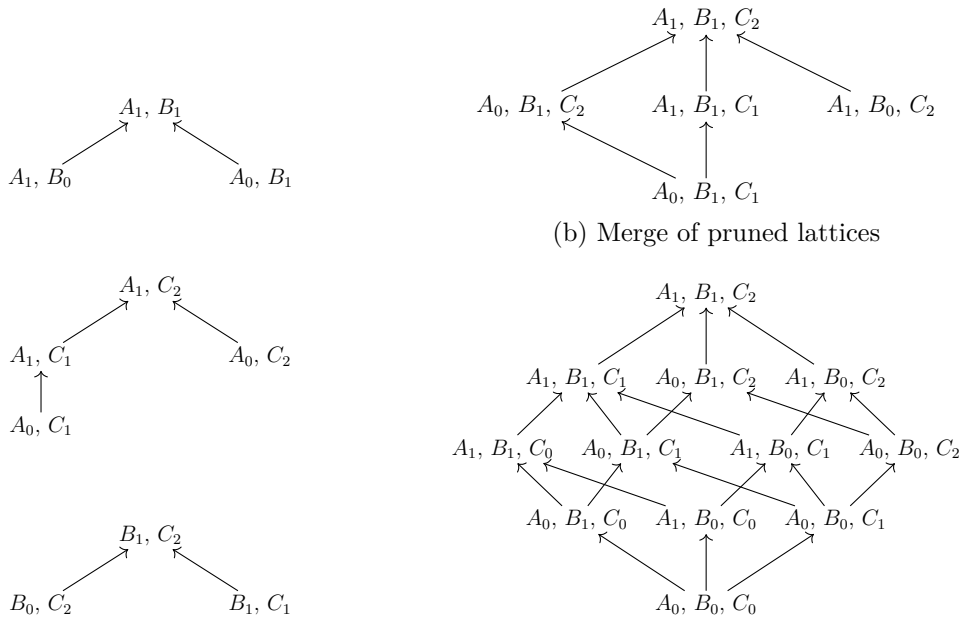
Another algorithm, the *Datafly* algorithm, was also presented by L. Sweeney [82] based on her previous works. Instead of focusing on an optimal solution, she proposed to use a heuristic. The Datafly algorithm computes frequencies of quasi-identifier attributes. Then it generalizes attributes which have the most distinct values until no value has less than k occurrences. This algorithm produces a lot of information loss. Together with the Datafly heuristic, L. Sweeney also proposed the $\mu-$*Argus* algorithm. It is a supervised $k-$anonymity algorithm which is inspired by the Datafly heuristic. It generalizes any attributes values with a frequency lower than k and then stores all quasi-identifiers which do not appear k times as *isolated records*. The data holder can then choose to remove these isolated records or to generalize them to produce new equivalence classes. Since the data holder chooses how to generalize isolated records, it can generate a big amount of information loss.

The *Bottom-up* algorithm, presented by K. Wang et al. [90], is a heuristic based on taxonomy trees (also called *Taxonomy Encoded Anonymity index*).

The taxonomy tree of an attribute a is a tree whose leaves are the different values of a in the dataset and the parent of each nodes is the generalization of the node. Their heuristic works as follows: it first creates a list of possible generalizations which modify the lowest cardinality of equivalence classes, and call them *critical* generalizations. Then it chooses one of the critical generalizations generating the lowest information loss. For example, let e be the only equivalence class containing one record (*i.e.* being the smallest equivalence class). The heuristic builds the list $\ell = \{g_a, g_b, g_c\}$ with $\forall g_i \in \ell$ applying g_i increases the cardinality of e. The generalization chosen will be the one minimizing the information loss. For instance, if a generalized record does not share the same quasi-identifier with any other record, increasing its equivalence class cardinality consists in generalizing this record to make it share the same quasi-identifier of at least another record. The heuristic is called *Bottom-up* due to the fact that it begins from the bottom of taxonomy trees (*i.e.* leaves nodes) and climb up with generalization to the top of trees (*i.e.* the root) so long as the anonymised solution does not satisfy the $k-$anonymity constraint. It displays feasible processing time when used with few attributes but explodes when anonymising a dataset with a large number of attributes (due to the fact that it lists all generalizations even if many are pruned by the *critical generalizations only* strategy).

In contrast to the Bottom-up heuristic, the *Top-Down Specialization* heuristic, presented by B. C. M. Fung [28], starts from the root and goes towards the leaves. The algorithm begins with all values completely generalized and de-generalizes (*i.e.* specializes) attributes while $k-$anonymity is satisfied. To identify in which order attributes are de-generalized, it makes use of a ratio (*i.e.* called a *score*) of an information loss metric over an anonymity metric (based on the number of each attribute's values before and after specialization). The Top-Down Specialization heuristic shows more feasibility than the Bottom-up algorithm.

The KACA algorithm was presented by J. Li et al. [57]. It is based on clustering techniques. It first considers all tuples as a one cardinality equivalence class and merges equivalence classes two by two until $k-$anonymity is reached. To select the two equivalence classes to merge, it takes a random one which has a cardinality lower than k and merges it with the equivalence class generating the lowest distortion. The distortion is a sum of predefined weights of generalizations needed to make that merge possible then amplified (*i.e.* with a product) by the cardinality of each equivalence class. This heuristic creates a lot of information loss and has high resources consumption.



(a) Pruned lattices for attribute's couples A, B and C

(b) Merge of pruned lattices

(c) Complete generalization lattice

Figure 3.2: Comparison of pruned lattices merge vs complete generalization lattice

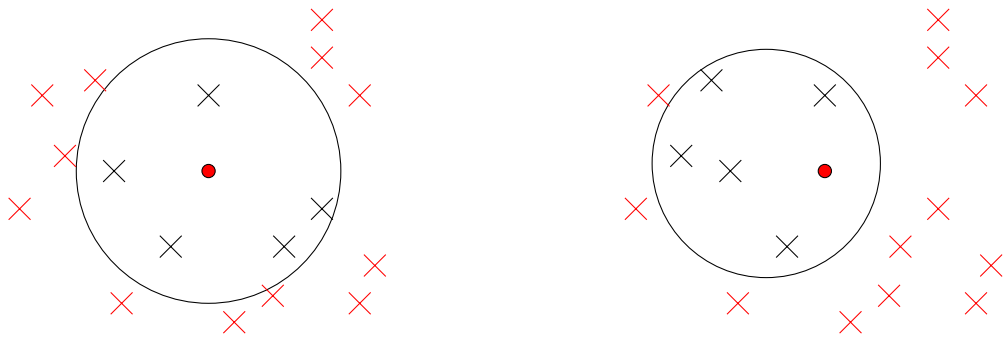
Some papers have presented improvement of the KACA algorithm such as the *Top-Down KACA* [98] which uses the Top-down Specialization heuristics to partition the dataset then uses the KACA heuristic on those partitions. The high information loss generated by the KACA algorithm is due to the fact that it constructs clusters which have too many tuples. A. Gionis et al. proposed a fix to that problem by moving records from the edge of big clusters to another cluster [38].

K.LeFevre et al. proposed the *Incognito* heuristic [53]. It is a lattice-based heuristic which improves the MinGen heuristic. Possible generalizations can be seen as a lattice with the upper bound as a fully generalized dataset and lower bound as original dataset. The Incognito heuristic works as follows : it creates a generalization lattice for a subset of quasi-identifier attributes (*e.g.* selecting only two attributes for example). It then prunes the lattice removing nodes which do not satisfy the $k-$anonymity requirement (*i.e.* for the targeted attributes). When it has generated different lattices for quasi-identifier attributes subsets, it merges lattices into one and again prunes

nodes which do not satisfy $k-$anonymity. This heuristics iterates on the merging of lattices and offers a better feasibility than MinGen. Since the generation of lattices takes a large amount of memory resources, K.LeFevre et al. use databases to save lattices. This induces a high processing time cost due to I/O operation on hard disk which are much slower than random access memory. Figure 3.2 shows the merging of pruned lattices vs unpruned lattice. This figure represents the generalization of three attributes A, B and C with X_i the ith generalization of attribute X. Attributes A and B can be generalized only once (*e.g.* the sex attribute) and the C attribute can be generalized two times (*e.g.* different granularity of age intervals). Figure 3.2a shows the three attribute couple lattice generalizations pruned – *e.g.* having no generalization over A and B does not satisfy $k-$anonymity since node A_0, B_0 has been pruned so A_0, B_0, C_i does not appear. Figure 3.2b shows the merge of the three lattices which is relatively less impressive than the complete lattice representing all possible generalizations shown by figure 3.2c.

K. LeFevre et al. also presented a new partition-based algorithm called *Mondrian* [54]. It is an algorithm which starts with a unique equivalence class containing all records from the database. We can see it as a top-down heuristic which cuts big equivalence classes in half on the widest attribute. So, the first equivalence class is partitioned into two equivalence classes on an attribute A. The intersection of the two equivalence classes is empty. Each equivalence class is then partitioned again into two equivalence classes. The process continues until each equivalence class has a size lower than $2 \cdot k$. It is a divide-and-conquer strategy which shows a high scalability due to its complexity in $O(n \cdot \log n)$. It also generates a high information loss quantity due to the fact that equivalence classes cardinality are almost always greater than k (*i.e.* between k and $2 \cdot k$).

J. Domingo-Ferrer and V. Torra proposed the MDAV algorithm based on micro-aggregation [21]. The concept of micro-aggregation is to build equivalence classes then choosing symbolic attribute values (*i.e.* an aggregation) to represent equivalences classes. It uses mean values to represent continuous attributes and median to represent categorical attributes. To make equivalence classes, it uses a centroid based clustering technique. It begins by computing the centroid c of the dataset then designates two centroids opposed to c and the farthest from c. It completes the two clusters (*i.e.* equivalence classes) by adding nearest records to the designated centroids. When clusters have enough records (*i.e.* more than k), they are published and the process is reiterated until all records are published. The MDAV



(a) MDAV (b) Greedy $k-$member clustering

Figure 3.3: Difference in the creation of equivalence class between MDAV and Greedy $k-$member clustering heuristics

heuristic shows good processing time and also low information loss.

Another heuristic similar to the MDAV algorithm has been presented by J.-W. Byun et al. and is called the $k-$Member Greedy. Instead of computing a centroid of the dataset, it takes a random record and builds a cluster from its farthest record r. Also it does not assign r as the centroid of the cluster. It takes an information loss metric which can give the information loss of a cluster and puts records which generate the least information loss to this cluster. When the cluster has enough elements, it is published and the records from this equivalence class are removed from the dataset. This process is reiterated until all records are published. It generates less information loss than the MDAV heuristic but also has greater processing time. It is due to the fact that the MDAV heuristic compares any record to the centroid which gives a complexity in $O(n^2)$ for the MDAV heuristic while $k-$Member Greedy computes the information loss of the cluster using each record, which for most information loss metrics has a complexity in $O(|c|)$ with c the cluster (*i.e.* the k parameter of the $k-$anonymity). $k-$Member Greedy anonymises a dataset in $O(n^2 \cdot k)$ operations. Figure 3.3 illustrates the difference between MDAV and $k-$Member Greedy heuristics. The red dots represent the first element placed in the cluster (*i.e.* chosen as the cluster's centroid for the MDAV heuristic).

The *Grading, Centering Clustering and generalization* (*GCCG*) algorithm presented by S. Ni et al. [71] is also a centroid-based clustering heuristic to achieve $k-$anonymity. Like the MDAV heuristic, it selects a centroid to build

equivalence classes by adding near centroid records in it. Instead of selecting the furthest record from the dataset centroid as equivalence class centroid, it sorts the dataset by scoring all entries. The record with the highest score is then chosen as the next centroid. The score of an attribute for a record is the ratio of this value over the attribute domain for categorical value and the proportion of the ratio to k for continuous attributes. This aims to begin with the densest equivalence classes and finish with equivalence classes composed of outliers.

It is important to note that all the algorithms described here were built to produce $k-$anonymous datasets with the assumption of a uniform k. They were not built with the idea of a personalised security parameter k (possibly one per record), which is an approach that we investigate in this thesis.

3.2.3 Related work on *differential privacy*

PPDP methods have a common weakness, publishing multiple anonymised datasets permits to infer individual's information. It is due to the fact that databases are entities which evolve over time. Anonymising a database in different points in time would mostly generate two different anonymised tables. Crossing those tables would permit to reduce record generalizations, inducing a privacy loss. The principal weaknesses of partition-based concepts come from the fact that this privacy loss is difficult to evaluate. Another approach, differing from the partition-based approaches and which is able to evaluate privacy loss of multiple anonymisations, is *differential privacy*.

In 2006, C. Dwork proposed a definition of $\epsilon-$differential privacy [22] to build a new family of anonymisation concepts. The idea of differential privacy is to ensure that the result of a query on a dataset is indistinguishable from the result on a neighbor dataset (*i.e.* a dataset which differs from at most one record).

Definition 3 (Differential privacy (from [22]))**.** A randomized function $\mathcal{K}$ ensures $\epsilon-$differential privacy if for all $\mathcal{D}_1, \mathcal{D}_2$ differing from at most one entry, and all $S \subseteq Range(\mathcal{K})$

$$\frac{Pr\left[\mathcal{K}(\mathcal{D}_1) = S\right]}{Pr\left[\mathcal{K}(\mathcal{D}_2) = S\right]} \leq e^{\epsilon} \tag{3.2}$$

C. Dwork also proposed a mechanism based on the addition of (Laplacian) noise to the result of a query. Figure 3.4 illustrates this differentially private

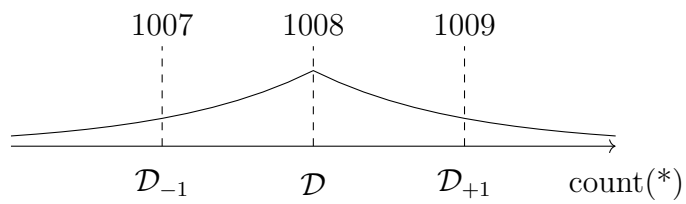


Figure 3.4: Representation of the differentially private mechanism based on the Laplacian noise (*i.e.* $Lap(|\mathcal{D}|, \epsilon = 1)$)

mechanism. The addition of Laplacian noise consists in selecting a random number from the Laplacian distribution and adding it to the result of a query. In the example, the Laplacian distribution is represented by the curve centered on the dataset $\mathcal{D}$. Assume a querier asks for a computation of the cardinality of the dataset $\mathcal{D}$ (*i.e.* called a *count(*)*). The dataset $\mathcal{D}_{-1}$ represent a neighbor dataset of $\mathcal{D}$ with a record less and the dataset $\mathcal{D}_{+1}$ is the neighbor with a record more. Since $|\mathcal{D}|$ is equal to 1008, the Laplacian noise is centered on 1008 and a random number is selected from this distribution as result of the query. Using a Laplacian distribution with $\frac{1}{\epsilon}$ as scale parameter, grants an $\epsilon-$differential privacy protection to this function. The mechanism used on figure 3.4 ensures $1-$differential privacy for the $count(*)$ operation. A lower ϵ would give better privacy (*i.e.* consisting of flattening the Laplacian distribution curve), but would worsen the quality of the result (*i.e.* error margins would be greater).

When computing a different function (*i.e.* instead of cardinality), the same Laplacian noise would not give the same security. This is due to the sensitivity of a function (*i.e.* called L_1-*sensitivity*).

Definition 4 (L_1-sensitivity of a function (from [22]))**.** For $f : \mathcal{D} \rightarrow \mathcal{R}^d$, with d a positive integer. The L_1-sensitivity of a function f, noted Δf, is defined as follows:

$$\Delta f = \underset{\mathcal{D}_1, \mathcal{D}_2}{max} \|\mathcal{D}_1 - \mathcal{D}_2\|_1 \tag{3.3}$$

with $\mathcal{D}_1$ and $\mathcal{D}_2$ two neighbor datasets and $\|\cdot\|_1$ denoting the norm ℓ_1.

For example, the L_1-sensitivity of the cardinality computation is $\Delta f = 1$ by definition. The Laplacian noise with b as scale parameter ensures a $\frac{\Delta f}{b}-$differential privacy. C. Dwork also showed that the composition of differentially private mechanisms weakened the privacy of individuals, which lead

to the introduction of the concept of privacy budget. The *privacy budget* refers to the preciseness of the information allowed to leak (*i.e.* represented by ϵ). Using a differentially private mechanism with ϵ_1 as security parameter consumes some of the privacy budget leaving $\epsilon - \epsilon_1$ budget. This is due to the sequential composition theorem presented by F. McSherry and K. Talwar [63] and recalled by the theorem 5.

Theorem 5 (Sequential Composition (from [63])). *Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ be a set of privacy mechanisms each providing ϵ_i-differential privacy guarantee, the sequential application of $\mathcal{M}_i$ on a dataset $\mathcal{D}$ provides $(\sum_{i=1}^{n} \epsilon_i)-$differential privacy guarantee.*

In our example, the querier asks for a computation of cardinality. If the privacy budget allowed is $\epsilon = 1$ then adding a Laplacian noise with scale parameter $b = 1$ would totally consume the privacy budget leaving no more possibility to run other queries. In practice, ϵ is partitioned in such a way to allow a predefined number of queries. Once the budget is consumed, it is no longer possible to ask any queries. Another way to view mutiple queries is that one is able to evalute the risk linked to their publication by summing the ϵ_i of each release.

From the differential privacy definition 3 a vast amount of concepts have emerged to propose different ways to apply differential privacy [100] by reducing the amount of noise added to the result or reducing how much privacy budget is consumed. Most of differential privacy mechanism propositions stick with a relaxation of the original definition of differential privacy.

Using exponential noise (*e.g.* Laplacian noise, Gaussian noise) to achieve differential privacy shows some weaknesses. It gives the same noise to low cardinality databases and high cardinality databases with a different impact on the two. Because of the privacy budget consumption the exponential noise mechanism also behaves badly with large set of queries. That is why R. Aaron and R. Tim proposed a system to "*factorize*" similar queries [76]. Their idea is to find queries that can be computed (or at least estimated) with the result of another query. For example, let us assume that a querier wants to compute the query Q_1 counting a number of diseases in the interval $[a, b]$ of age, and a query Q_2 similar to Q_1 with a different granularity of age $[a, c]$ with $b < c$. An estimation of the result r_2 of Q_2 could be an operation on the result r_1 of Q_1 such as $r_2 = r_1 \cdot \frac{b-a}{c-a}$. Only r_1 would be noisy having the effect to decrease the number of queries and so, reducing the privacy budget

consumption. Then, r_2 becomes an estimation of the query Q_2 based on the result of Q_1.

Instead of reducing the number of queries to decrease the privacy consumption, another way is to partition the dataset and apply differential privacy on the partitions. Since the partitions are distinct subsets from the dataset, another composition theorem can be used, the *parallel composition* theorem. The parallel composition theorem has been presented by F.D. McSherry [64] and is recalled by theorem 6.

Theorem 6 (Parallel Composition (from [64])). *Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ be a set of privacy mechanisms each providing ϵ_i-differential privacy guarantee, and $\mathcal{D}_1, \ldots, \mathcal{D}_n$ be disjoint subsets of the dataset $\mathcal{D}$, the application of $\mathcal{M}_i(\mathcal{D}_i)$ with $i = 1 \ldots n$ provides $max_{i=1}^{n}(\epsilon_i)-$differential privacy guarantee.*

F.D. McSherry proposed, with the PINQ platform, to give analysts the possibility to specify how partitions are made (*e.g.* to make histogram studies). Partitioning the dataset to publish histograms feels natural when dealing with range queries and since the histograms are built on disjoint subsets, the parallel composition theorem can be used. However partitioning the dataset by some variable attribute such as age, would give many partitions with a more or less large cardinality. As said before, the exponential mechanism to achieve differential privacy is not scalable (leading to high noise in low cardinality group). J. Xu et al. propose a mechanism to restructure the dataset merging some bins together [97] (*i.e.* in histograms, a bin or bucket is an interval in which values are represented by the same bar). They showed two propositions of algorithms. One merging bins after separately adding noise on them and the other merging bins before adding noise.

Managing small datasets

Because of the non adaptation of the Laplacian noise to small values, answering noisy queries with a small result leads to a large relative error. X. Xiao et al. proposed an iterative mechanism which reduces the amount of noise added to small answers, they call it *iReduct* [96]. The idea of their algorithm is to partition the privacy budget while keeping some in reserve. For a given set of queries, the algorithm computes the result of queries and applies the noise (*i.e.* using Laplacian noise) on them. Since it has kept some privacy budget in reserve, the algorithm selects queries with low results and rescales the privacy budget allocated to them by using the privacy budget in

reserve. For instance, let $\mathcal{Q} = \{Q_1, \ldots, Q_n\}$ be a set of n queries and ϵ be the privacy budget allowed for the answer. This privacy budget is partitioned into $n+1$ parts $\{\epsilon_1, \ldots, \epsilon_n\}$ being privacy budget allocated for queries and ϵ_r being the privacy budget in reserve. For $\epsilon_i \in \{\epsilon_1, \ldots, \epsilon_n\}$, ϵ_i should be lower than ϵ_r to allow multiple re-scaling of the privacy budget. A Laplacian noise ensuring an ϵ_i-differential privacy is applied on the answer of query Q_i (the noisy answer is noted Q_i^*). The algorithm selects a subset $\mathcal{Q}' \subset \mathcal{Q}$ of query answers with lowest results such as $\forall q' \in \mathcal{Q}', \forall q \in \mathcal{Q}/\mathcal{Q}', q'^* < q^*$. The privacy budget in reserve ϵ_r is then partitioned and the parts are added to the privacy budget allocated to queries in subset $\mathcal{Q}'$. We call this process noise re-scaling and the partionning of ϵ_r does not necessarily consume the whole privacy budget in reserve and the rest will be used in next iterations. The noisy result of each selected query is re-computed again with the new privacy budget. In next iterations, previously selected queries should not be selected again for noise re-scaling. This process continues until the whole privacy budget in reserve is totally consumed. It is important to note that consuming the whole privacy budget in the first iteration consists in applying a simple Laplacian noise. To resume, the iReduct algorithm increases the noise of high value results and decreases the noise of small value results.

All previous methods were based on the use of Laplacian noise. Another differentially private mechanism has been proposed by F. McSherry and K. Talwar at the same time of the sequential composition theorem [63]. They showed how to design a differentially private mechanism by proposing a general approach called the *Exponential Mechanism*. Here is the definition of the exponential mechanism:

Definition 7 (Exponential mechanism (from [63])). For any function $f : (\mathcal{D}^n, \mathcal{R}) \rightarrow \mathbb{R}$ and base measure μ over $\mathcal{R}$, the exponential mechanism (noted $\mathcal{E}_f^\epsilon(d)$) gives $(2 \cdot \epsilon \cdot \Delta f)-$differential privacy for the dataset d:

$$\begin{aligned} \mathcal{E}_f^\epsilon(d) :=& \textit{Choose } r \textit{ with a probability proportional to} \\ & \exp\left(\epsilon \cdot f(d, r)\right) \cdot \mu(r) \end{aligned} \tag{3.4}$$

with $\mathcal{R}$ being a range of output.

The goal of exponential mechanism is to maximize $f(d, r)$ while ensuring differential privacy. The function f can be seen as a utility measure for query q and so, the $r \in \mathcal{R} = Range(q)$ solution has exponentially more chance to

be the one with the highest utility. Note that the Laplacian noise mechanism can be captured with the exponential mechanism.

All previous methods aim to answer a set of queries while ensuring differential privacy. Since those methods are limited to a given set of queries, other studies aim to make a synthetic dataset built upon the original one while ensuring differential privacy. For example, N. Mohammed et al. [68] suggest to use partition-based anonymity methods (*e.g.* TopDown specialization) to build partitions of the dataset, then add Laplacian noise to the count of each partition. Note that they do not try to reach $k-$anonymity. Following a given specialization number s, their algorithm, *DiffGen*, specializes an attribute. This attribute is selected with a probability based on the exponential mechanism and a score. Their score is based on two metrics, one is the information gain, the other is a classifier metric that aims to keep the highest number of records with the same class[1] in a group (*i.e.* an equivalence class). When s specializations have been done, the DiffGen algorithm publishes the count of each group after adding some Laplacian noise. As specified, before, the Laplacian noise does not get along with small numbers and so, they exhibit the fact that a too high number of specializations reduces the cardinality of partitions increasing the impact of the noise. It also decreases the accuracy of the classification metric (*i.e.* losing some information on the classifier attribute).

J. Zhang et al. presented the *PrivBayes* algorithm based on a Bayesian network. This algorithm synthesizes a dataset from a noisy Bayesian network. It works as follows: first, PrivBayes constructs a Bayesian network using the differentially private exponential mechanism. To build the Bayesian network, it computes the mutual information I of attribute associations (X, Π) with Π as the parent of X as follows :

$$I(X, \Pi) = \sum_{x} \sum_{\pi} Pr\left[X = x, \Pi = \pi\right] \log \frac{Pr\left[X = x, \Pi = \pi\right]}{Pr\left[X = x\right] \cdot Pr\left[\Pi = \pi\right]}$$

The exponential mechanism is applied using mutual information as the utility function (*i.e.* f in the definition of exponential mechanism) to select the pair (X, Π) as edge on the Bayesian network. When a pair is identified, the

[1] A class is an attribute which is important to keep the less generalized. For example, in the adult dataset from the census bureau database [59] the attribute income is a boolean which indicates if a user has an income lower or greater than $50K$. It is a classification of individuals in the data set and so, it is called a classifier attribute.

algorithm removes X from its domain (*i.e.* a same attribute cannot be selected twice as X value). Then, it computes the distribution of each pair (X, Π) of the Bayesian network and adds a Laplacian noise to those distributions. Finally, the algorithm builds a synthetic dataset following the noisy Bayesian network built firstly and the noisy distributions built secondly.

3.2.4 Related works on personalisation of privacy-preserving methods

All previously presented concepts lack the possibility to empower users. Since the GDPR [75] tries to push user empowerment initiatives, giving the possibility to users to push their own perception of the risk is an important feature. Existing partition-based techniques are uniform, they focus on a fixed security parameter. It provides the same privacy guarantees to users, even if these users do not aim for the same privacy goal. This section presents studies which are directly related to the personalisation of privacy-preserving methods. Those methods aim to provide the possibility to use existing privacy preserving approaches with a variable security parameter providing differentiated protection to each user.

In 2005, B. Gedik et al. proposed a way for users to personalise the $k-$anonymity concept by specifying different k in a mobile network [35] called the *Clique-Cloak* algorithm. Their proposition focused on location based service system and real-time queries which induces a non measurable privacy loss due to the multiple $k-$anonymous datasets published. Their idea is based on a trusted third party server which is a bridge between users and the location based service (LBS) servers. Users specify bounds on the maximum degradation of their location (*i.e.* a box), the security parameter of the $k-$anonymity (*i.e.* the number of people in the box) and a constraint of time. The system creates a graph representing each user as a node. Two nodes are linked if they are each in the box of the other. To ensure that users privacy constraints are satisfied, the Clique-Cloak algorithm only sends cliques of size greater or equal to the maximum user constraint. Any "box" sent to the LBS ensures $max(k_i)-$anonymity with $max(k_i)$ the highest constraint from each user in the clique. This concept can work in populated cities but is much less efficient in low populated campaign (*i.e.* or with a much larger box).

Another algorithm, the Casper algorithm, has been proposed by M. F.

Mokbel et al. [69]. The idea of the Casper algorithm is to use a *pyramid* where each layer represents different granularities of the projection of users on a map. Pyramid layers are cut into cells of different sizes in function of the layer. The pyramid is kept up-to-date on a trusted third party server. Users send updates of their location and when a user wants to query the LBS server, the trusted server sends the user cell on the layer with sufficient users (*i.e.* greater than the user constraint).

The PrivacyGrid algorithm, presented by B. Bamba et al. [7], ensures personalized $k-$anonymity for LBSs and also proposed a way to ensure personalized $\ell-$diversity. Their idea is similar to the Casper algorithm but it does not keep layers of the pyramid up-to-date, it uses a map instead. It computes the smallest area which satisfies each user constraint and sends it to the LBS service.

These solutions are all based on a trusted third party which ensures user privacy constraints, and focus on location data. The application of these solutions aims at providing a service to users. The data quality is not a high priority since each user's mobile can sanitize answers from LBSs. They also focus on real-time queries (*i.e.* users sends a query and expects to get answers in a short lapse of time) which magnifies the weakness of privacy methods (*i.e.* repeating different anonymisations induces a loss of privacy).

Another way to personalise $k-$anonymity was first proposed by X. Xiao et al. [95] and consists in personalising the generalization of sensitive attributes. Users specify how their sensitive attributes will be shared (*e.g.* if users want to be in a group of similar diseases or on the contrary with distant diseases). A taxonomy tree for the sensitive attribute is needed. Users specify a node on the taxonomy tree (called a *guarding node*) to indicate they do not want to be associated to any leaf values of the sub-tree with guarding node as root. Other papers based on the $(\alpha, k)-$anonymity [92] were proposed by also personalizing how sensitive attributes are shared [78, 60]. The paper [60] adds the principle of guarding node proposed by X. Xiao et al. [95] to $\alpha-$deassociation. The $\alpha-$deassociation concept consists in specifying a frequency threshold α and a sensitive value s. This sensitive value will not be more frequent than α in each equivalence class (note that α should not be lower than the frequency of s on the whole dataset). The data holder specifies a threshold α_s for each sensitive value and users can specify a guarding node. On the other hand, the paper [78] proposes to users, a way of specifying the sensitivity of their sensitive attribute. Those solutions induce a loss of quality of the sensitive attribute and do not fit with PPDP

where there could be zero or multiple sensitive attributes.

On the other hand, ideas to personalise differential privacy were also proposed. Z. Jorgensen et al. proposed an algorithm based on a sampling mechanism [45]. Each user can alter the probability to participate to the query by modifying their epsilon. To ensure that all users receive at least minimal security, the privacy budget is partitioned. The first part is dedicated to the sampling mechanism (*i.e.* the personalisable part) and the second part is dedicated to another differentially private mechanism (*e.g.* Laplacian noise mechanism).

More recently, N. Li et al. proposed a new personalised differentially private mechanism [56]. The idea is to partition users by their personalized security parameter ϵ_i. A differentially private mechanism is applied to each partition with the security parameter common to users inside it. Due to the parallel composition theorem 6 their solution offers an $\epsilon-$differentially private measure with ϵ equal to the highest ϵ_i users have chosen. When dealing with personalized differential privacy, the general definition of differential privacy makes no sense due to the parallel composition theorem and so, they proposed another definition of personalized differential privacy which is based on $\epsilon-$differential privacy.

Geo-indistinguishability is a concept presented by Andrés M. E. et al. [5]. The concept adds a two dimensional Laplacian noise to users location. This mechanism can be illustrated by a circle on the map with the user position in. Users have the possibility to enlarge this circle which gives them a personalised differential privacy. Since it is an algorithm to hide each individual from LBSs, the overall information loss is not essential.

3.3 Privacy-Preserving in Interactive Query Systems

In previous sections, we presented existing PPDP methods, which allow the release of anonymised datasets. Except for differential privacy, the PPDP approach is not adapted for interactive queries. As their name suggests, interactive queries offer a way to interact directly with a database. The concept of private databases is emerging and a large amount of studies proposing different paradigms and systems to achieve privacy-preserving data computations have been done [16]. In this section we show different privacy-preserving

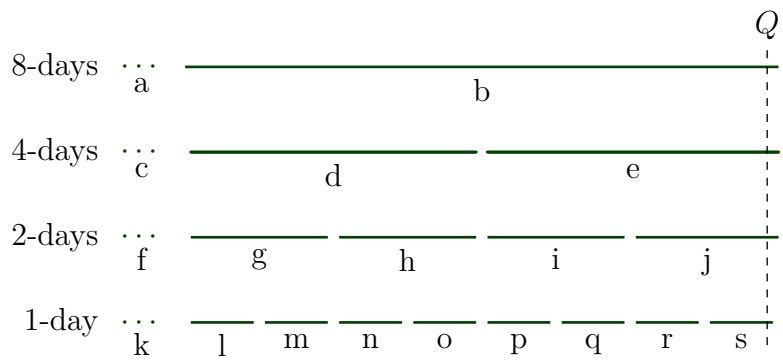


Figure 3.5: Partitions of the database by granularity of time (from [27])

systems in which a querier (*i.e.* someone posting a query) can interact with the database.

Like in PPDP methods, privacy-preserving can be ensured in interactive query systems by using differential privacy. M. Hardt and G. N. Rothblum introduce the *private multiplicative weights* (PMW) mechanism [41]. The database (or data universe) is viewed as a histogram with a weight on each bin (remember that bins are intervals on which bars of histogram are computed). The initial weights are equal to $\frac{1}{N}$ with N being the number of bins (*i.e.* a uniform distribution). When a query needs to be answered, PMW uses weights to compute the result and adds Laplacian noise to it. If the noisy result is too far from the real result (*i.e.* computed on the original database), weights of bins targeted by the query are adjusted to better approximate the real result. When another query needs to be computed, PMW checks if the previous noisy answer can be used to answer this new query (*i.e.* the previous answer is near the new noisy answer). If so, the previous answer is given and no privacy budget is consumed. Otherwise, weights are adjusted again. Note that the PMW mechanism can answer a large amount of queries.

The *Diffix* system [27] presented by P. Francis et al. can be considered as an SQL proxy. Their objective is that an analyst should be able to send an SQL query to the system. The Diffix system adds Gaussian noise on the result of each query. To prevent the privacy budget to be consumed, they use *sticky noise*. The PRNG (pseudo-random number generator) uses a seed to produce a sequence of random numbers. They call sticky noise the fact that the seed used by the PRNG is parameterised by the query (*i.e.* the seed is the hash of a concatenation of table column names, condition values, condition

operators and a secret salt). The Diffix system, also takes into account the evolution of the database through time. They cut the database in different granularities of time (see their example figure 3.5). In the figure, the $a, \ldots, s$ represent time intervals (*e.g.* day by day for 1−day layer) called *epochs*. Let Q be the query at a time t, since this query happens during the *epoch* s, the answer will be computed from the state of the database at the end of the day r (*i.e.* the state at the end of the *epoch* before the query). To ensure any modification of the database has a small impact on the noise added by previous security mechanism, the Diffix system adds noise with b, e, j and s *epochs* as seeds. This has the effect to reduce the modification of the noise when the database is updated and so, makes it difficult for an attacker to average out the noise.

Another approach uses fully homomorphic encryption. A homomorphic encryption is an encryption which allows computations on the cipher text without decrypting it. Those computations on cipher texts permit the arithmetic addition and multiplication of plain texts. For instance, let c_1 and c_2 be two cipher texts of respective plaint texts m_1 and m_2 encrypted by the well known Paillier encryption scheme [73]:

$$c_1 = (1+N)^{m_1} r_1^N \mod N^2 \tag{3.5}$$

with r_1 as a random number and N the product of two big prime numbers. The Paillier scheme is said to be additively homomorphic because of the fact that the product of cipher texts permits to achieve the cipher of plain texts addition:

$$\begin{aligned} c_1 \cdot c_2 &= (1+N)^{m_1+m_2}(r_1 \cdot r_2)^N \mod N^2 \\ &= Encrypt(m_1 + m_2) \end{aligned} \tag{3.6}$$

The difference between fully homomorphic encryption and non fully homomorphic encryption is that the fully homomorphic encryption can do both arithmetic operations (+ and ×, thus producing a ring morphism) when homomorphic encryption can only do one of them (thus producing a group morphism). In our example, the Paillier scheme can only produce the addition of plain texts without decrypting cipher texts and so, is said to be additively homomorphic. C. Gentry proposed the first fully homomorphic encryption [36] scheme based on ideal lattices. Y. Gahi et al. proposed a secure database management system (DBMS) [30] using this fully homomorphic encryption scheme. Their idea is that a client can make use of a cloud

provider to delegate computations over its data. The problem is that the client does not want the cloud provider to infer data the client sent to it. The client sends encrypted data using the Gentry encryption system and then asks the cloud provider to make the arithmetic additions and multiplications needed. They show that a multiplication between two 16-bit integers took 23 minutes on their 1.7GHz processor and that an SQL select operation needed more than 300k multiplications for a database containing 10 records. Obviously, this explains why despite many improvements, fully homomorphic encryption systems are still not suitable today to delegate computations on encrypted data to a cloud provider.

On the other hand, the *CrypDB* system [74], presented by R.A. Popa, uses different homomorphic encryption schemes (*i.e.* non fully homomorphic) which are much more feasible. They introduce the term of *onions* which design an encapsulation of encryption schemes. An onion has multiple layers of encryption. A first layer (RND) consists to use a block cipher such as AES in CBC mode. The DBMS should be able to decrypt this layer. However other layers should not be decrypted by the DBMS. Each encryption system permits the use of different operations such as *join*, *equality*, *additions*, ... Since different encryption systems may be inconsistent with others the CrypDB system replicates data in different onions. Distinct onions aim to be able to compute distinct operations. The use of onions may not be feasible due to the massive replication of data to allow the different operations.

Even if non fully homomorphic encryption systems offer more feasibility, they have a significant cost. S. Bajaj and R. Sion have presented *TrustedDB* [6] which is based on a secure cryptographic coprocessor (SCPU). The TrustedDB system is made of an untrusted server hosting the database. They consider the fact that some personal information may be private and other may be public. The untrusted server contains a SCPU which possesses couples of private, public keys and a *symmetric* key. The SCPU contains different applications organised in layers (*e.g.* a query parser, query dispatch, ...). Couples of private, public keys are used to provably assert any procedure of the SCPU to remote parties (*i.e.* it ensure remote parties that the SCPU does what it should). Private data is encrypted with the symmetric key and stored together with plain text public data. Queries received by the host server are forwarded to the SCPU which decrypts the queries and dispatches queries to a deeper layer (*i.e.* the module which compute queries) for private queries (*i.e.* requiring computations on private data) or out of the SCPU for public queries. Query answers are encrypted with the client public key

by the SCPU to ensure only the client can understand the result. To sum up, TrustedDB is a system where an untrusted host server supervised by a trusted SCPU which can make secure computations.

Q.C. To et al. proposed a novel approach with the desire to avoid the use of trusted third party. They proposed the *Trusted Cells* architecture [85] where personal data is fully distributed and managed under user's control thanks to their personal devices. The objective of the Trusted Cells architecture is to use a large amount of trusted devices with low resources. Their characteristics are similar to smart-cards and to go beyond this limit, an untrusted cloud is used. They also proposed a protocol to compute SQL queries while ensuring the cloud does not infer any personal information.

3.4 Data Quality and Information Loss Metrics

The goal of personalised privacy is to increase individuals control over their own data and the trust they have on data controllers. However, such an approach may improve the data utility of a category of persons while worsening the impact of anonymisation on the data utility for another category of persons. This is why it is important to be able to measure the impact on data utility. This section aims to provide an overview of the different metrics of data quality and information loss.

Since the goal of partition-based privacy-preserving techniques is to group similar data together, anonymisation can be taken such as a clustering problem. In PPDP and clustering a vast amount of metrics to compute information loss (*i.e.* dissimilarity) or to evaluate data quality (*i.e.* similarity) exist. This section shows a study of different metrics used in clustering problems and in partition-based PPDP techniques to evaluate the quality (*resp.* the information loss) of a solution or the similarity (*resp.* dissimilarity) of different records.

3.4.1 Metrics in Clustering Techniques

The goal of clustering algorithms is to optimize a function while grouping similar records together. One of the most well known functions to optimize is to minimize the maximum diameter of clusters [39] (*i.e.* the diameter of

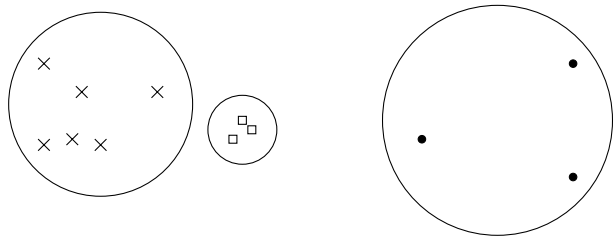


Figure 3.6: Minimizing the diameter

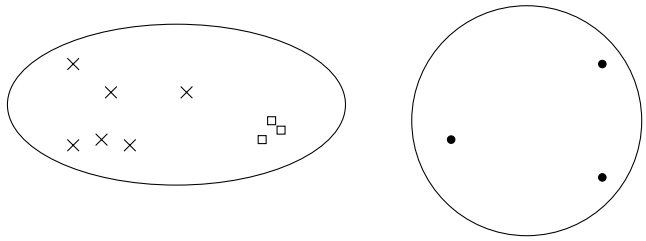


Figure 3.7: Maximizing the space between clusters

a cluster is denoted by the maximum distance between two objects of the cluster). Figure 3.6 illustrates the disadvantage of this metric when used to make 3−anonymity. Due to the fact that outliers (*i.e.* people which are not similar to others), symbolized by dots, generates a high cluster diameter, denser region may generate lower clusters. In figure 3.6, the region with crosses is much denser than the cluster grouping dots, but the metric does not make any difference between grouping all crosses in one cluster or separating them in two clusters (due to the fact that the diameter of one cluster would not be greater than the diameter of outliers cluster).

Another metric is based on the distance between clusters. The goal of this metric is to maximize the minimum distance (*i.e.* the split) between clusters (*i.e.* single-link clustering). The minimum distance between two clusters is the lowest distance between two objects from each cluster. The problem of this metric is that close objects may be in the same cluster because their distance is too small. The figure 3.7 shows this problem. First, as the diameter metric, crosses are merged in the same cluster because of their proximity. Second, squares are merged with crosses because crosses are much closer than dots. Maximizing the split criterion leads to create the lowest number of clusters.

All these problems come from the fact that most of clustering algorithms

need to know the precise number of clusters to obtain in advance. These algorithms try to optimize previous metrics (*i.e.* also called optimization criteria) by keeping a given number of clusters. This is contrary to the goal of $k-$anonymity which tries to create the highest number of clusters (*i.e.* equivalence classes) with a lowest cardinal (*i.e.* with k as lower bound) in general.

Another well known metric is the *Within Cluster Sum of Squares* (*i.e. WCSS*). This metric considers the distance of each record in a cluster to the centroid of this cluster. Equation 3.7 shows how this metric is computed:

$$\sum_{i=1}^{|\mathcal{C}|} \sum_{j \in c_i} dist(j, \mu_i)^2 \tag{3.7}$$

with c_i the ith cluster in the set of clusters $\mathcal{C}$ and μ_i the centroid of the cluster c_i. This metric is used by the well known $K-$means algorithm [62]. This metric may be hard to compute for categorical attributes since the mean between categorical attribute must be defined beforehand. One idea for example is to follow the approach of J. Domingo-Ferrer and V. Torra [21], who define the mean of categorical values as the most frequent value. To overcome this weakness, it is possible to compute the *Within-Cluster Sum of Dissimilarities* (*i.e. WCSD*) which is the sum of pairwise distances in a cluster as illustrated by the following equation 3.8:

$$\sum_{i=1}^{|\mathcal{C}|} \frac{1}{2} \sum_{x,y \in c_i} dist(x, y) \tag{3.8}$$

3.4.2 Quality Metrics for Anonymisation Techniques

Metrics of PPDP methods share some similarities with clustering optimization criteria such as the use of Manhattan or Euclidian distances. They also are dissimilar since their goal is different. We can distinguish two kinds of metrics in PPDP:

- metrics that evaluate the quality of a solution like the discernability metric [8], the normalized average equivalence class size metric [54] or information loss [14]. These metrics aim to provide information on the quality of a $k-$anonymous solution.

- metrics that evaluate the information loss on a specific attribute type (*e.g.* numerical, categorical). A vast amount of metrics exist for numerical attributes (*e.g.* based on the distance of anonymised value from the original, on the correlation or on the covariance [18]). Metrics about information loss on categorical attributes are fewer (*e.g.* based on entropy, on contingency tables or on taxonomy trees).

The normalized average equivalence class size metric C_{AVG} was proposed by K. LeFevre et al. [54]:

$$C_{AVG}(\mathcal{D}^*) = \frac{|\mathcal{D}|}{|\mathcal{E}| \cdot k} \tag{3.9}$$

with $\mathcal{D}^*$ the anonymised version of dataset $\mathcal{D}$ and $\mathcal{E}$ the set of equivalence classes. This metric aims to show the distance from a *quasi-optimal* solution (*i.e.* a solution with only equivalence class of cardinality equal to k) and the anonymised solution tested. This metric considers the suppression of a record equivalent to putting a record to an equivalence class of cardinality greater or equal to k.

The discernability metric DM has been proposed by R. J. Bayardo and R. Agrawal [8]:

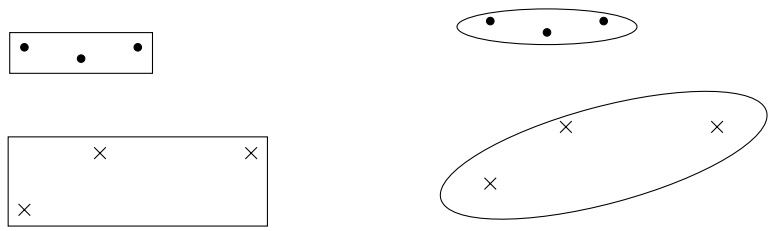
$$DM(\mathcal{D}^*) = \sum_{E \in \mathcal{E} \ s.t. |E| \geq k} |E|^2 + \sum_{E \in \mathcal{E} \ s.t. |E| < k} |E| \cdot |\mathcal{D}| \tag{3.10}$$

The goal of this metric is to minimize the size of each equivalence class and to give a penalty to equivalence classes which do not satisfy the k parameter. The problem of this metric is that it does not consider the distance between records which can lead to group records with high dissimilarity.

A metric based on the surface of an equivalence class was proposed by J.-W. Byun et al. [14]:

$$IL(e) = |e| \cdot \left(\sum_A \frac{dist_A(e)}{dist_A(\mathcal{D})} \right) \tag{3.11}$$

with $dist_A(e)$ the greatest distance on the attribute A between two records on the equivalence class e. This metric aims to evaluate the *surface* of equivalence classes as rectangles (*i.e.* each side represents the domain of the equivalence class on an attribute). Figure 3.8a illustrates the metric. The information loss of the anonymised solution consists in summing the information



(a) Surface-based information loss

(b) Diameter-based information loss

Figure 3.8: Differences between SBIL and DBIL metrics

of each equivalence class. We will call it $SBIL$ (Surface Based Information Loss) in the rest of this paper to avoid misunderstanding with information loss (IL).

The last metric, $SBIL$, uses computations of information loss on attributes. It is a sum of information loss over each attribute. We propose another information loss metric which is based on clustering metrics. We use the diameter of an equivalence class to evaluate how much information loss is generated by the anonymisation and call it the $DBIL$ metric (*i.e.* diameter-based information loss). The advantage of such a metric is that it just needs the distance between records to be computed (which is a constraint of clustering techniques). The $DBIL$ metric is defined by the following equation 3.12:

$$DBIL(\mathcal{D}^*) = \sum_{e \in \mathcal{E}} |e| \cdot diameter(e) \tag{3.12}$$

Information loss of numerical attributes can be expressed by multiple metrics. J. Domingo-Ferrer et al. [18] proposed a set of metrics to compute the information loss based on the distance of anonymised values from the original, on the Pearson correlation or on the covariance. Those metrics must be computed on the anonymised dataset and are made for modification of original values and not generalizing values by intervals. The most used metrics to evaluate information loss of continuous remains the Euclidean (or Euclidean squared to avoid the non-negligible cost of square root computation) and Manhattan distance.

To evaluate the information loss on categorical attributes, the use of Shannon entropy was first mentioned by P. Kooiman et al. [44]. An entropy based

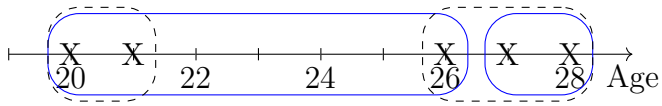


Figure 3.9: The EBIL metric weakness: the proximity of different attribute values

metric was then formalized by J. Domingo-Ferrer et al. [18]:

$$EBIL(A, \mathcal{D}^*) = \sum_{r \in \mathcal{D}^*} H(A|A'_r) \tag{3.13}$$

with A a variable in the original dataset and A'_r the value of this attribute for the record r in the anonymised dataset. The EBIL metrics was designed for categorical attributes and for global recoding. We can generalize this metric for local recoding with the following equation:

$$EBIL(A, \mathcal{D}^*) = \sum_{r \in \mathcal{D}^*} H(A|r \in e_r) \tag{3.14}$$

with e_r the equivalence class of the record r. The equation 3.14 can be computed by using the value distribution of A in each equivalence class. This metric aims to make equivalence classes with an unbalanced ratio of value for a given attribute (*i.e.* a value should appear much more than others in the same equivalence class). This is due to the fact that the probability a record r is associated to a sensitive value s when knowing r is in the equivalence class e can be computed as follows:

$$Pr\left[sensitive(r) = s \mid r \in e\right] = \frac{|\{i \in e \mid sensitive(i) = s\}|}{|e|} \tag{3.15}$$

with $sensitive(r)$ the sensitive value of r. Since the entropy is maximised when the probabilities are uniform, it is maximised when the frequencies of sensitive values in the equivalence class are uniform. The problem of the EBIL metric is that it does not take into account the proximity of different values. This explains why this solution is not used with continuous attributes. The figure 3.9 shows two sets of equivalence classes made in a one dimensional dataset with a continuous attribute. The EBIL metric does not identify any difference of information loss between the two sets of equivalence classes when the dashed set is intuitively better (*i.e.* when computing $2-$anonymity).

Another example would be the nationality categorical attribute : it would generate less information loss to generalize Spanish and French people together than with North Americans due to the proximity of Spain and France.

Another way to measure the information loss of generalized categorical attributes is the use of contingency tables. Some contingency tables are computed following given criteria on the original dataset and the anonymised one. The distances between the contingency tables are then used to show the information loss:

$$CTBIL(\mathcal{D}, \mathcal{D}^*) = \sum_{t \in T} \sum_{c \leq |t|} \mid x_c - x'_c \mid \tag{3.16}$$

with T the set of contingency table and x_c (*resp.* x'_c) the value of the cell c in the contingency table computed for $\mathcal{D}$ (*resp.* $\mathcal{D}^*$). This metric may need the computation of a high number of contingency tables and does not take into account the proximity of some value just like the EBIL metric.

Another information loss metric for categorical attribute is based on taxonomy trees. The idea of taxonomy trees is to specify how to generalize information. The figure 3.10 shows an example of a taxonomy tree (on the workclass attribute of the adult dataset). The leaf of the tree represents original values and each parent of a node represents the generalization of the node. The root of the tree is the fully generalized value. The information loss can be computed by using the height of the smallest subtree containing all values. A metric to capture that is used in [14], we will call it the taxonomy-based information loss (TBIL):

$$TBIL(A, e) = \frac{h(\Lambda_A(e))}{h(\mathcal{T}_A)} \tag{3.17}$$

with $\Lambda_A(e)$ the smallest subtree containing any value of the equivalence class e on the attribute A, $\mathcal{T}_A$ the taxonomy tree of the attribute A and $h(t)$ the height of the tree t. In the example figure 3.10, if an equivalence class e contains workclass values such as *State-gov* and *Local-gov*, the information loss would be computed as follows:

$$TBIL(workclass, e) = \frac{h(gov)}{h(\mathcal{T}_{workclass})} = \frac{1}{3} \tag{3.18}$$

The problem of this metric is that a taxonomy tree must be specified for each attribute to compute TBIL and the proximity between attributes values must also be specified.

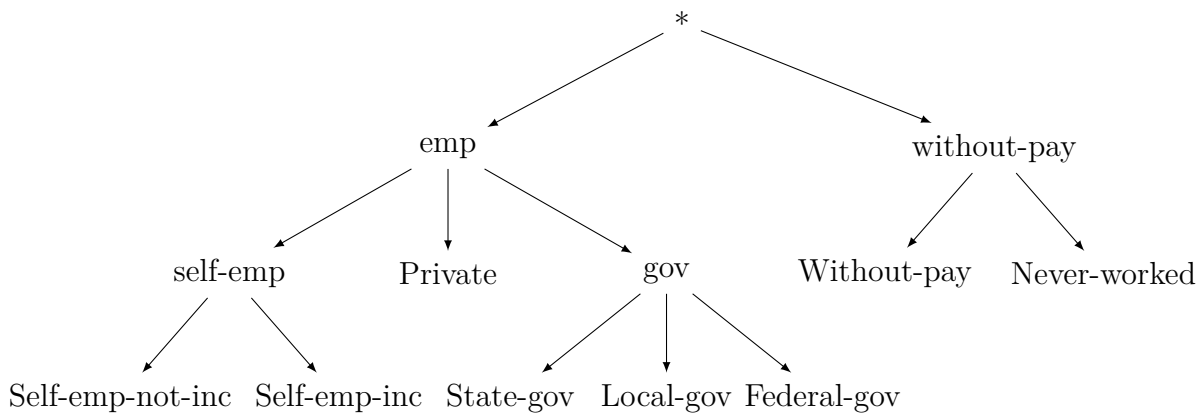


Figure 3.10: Workclass taxonomy tree

3.4.3 Disclosure risk metrics

It is important to note that even if previous metrics aim to measure the information loss induced by anonymisation methods, they do not measure how much anonymised individuals are. In their article [21], J. Domingo-Ferrer and V. Torra proposed to compute the quality of an anonymised dataset with the help of a score metric:

$$score(\mathcal{D}^*) = \frac{IL\,(\mathcal{D}^*) + DR\,(\mathcal{D}^*)}{2} \tag{3.19}$$

with $IL\,(\mathcal{D}^*)$ the information loss generated by the anonymisation of the dataset $\mathcal{D}$ and $DR\,(\mathcal{D}^*)$ a metric measuring the disclosure risk associate to the anonymisation of $\mathcal{D}$. They used a disclosure risk metric defined in another of their paper [20] based on the association of three different disclosure risk metrics. The three metrics were built for different objectives. The first metric is the *Distance Linkage Disclosure* (DLD) metric. The DLD metric can be computed when the anonymisation methods used modify original values (*i.e.* from $\mathcal{D}$) to others (also called masking). The DLP represents the proportion of masked values that can be linked to their original value. A link appear when the nearest value v from the masked value v' in the original dataset really is the original value of v' (*i.e.* no other masked value are nearest to v than v'). The second metric is the *Probabilistic Linkage Disclosure risk* (PLD) metric. The PLD metric is computed by making pairs of original and masked values using linear sum assignment problem. The percentage of

X	Y	$CDR_X(Y)$
Condition	Age*	0.25
Condition	Age	0.85
Age	Age*	0.54

Table 3.5: Value of CDR metric on table 3.3 using different set of attributes

correct pairs represent the value of PLD metric. While linear sum assignment problem can be used with masking methods, generalization methods can also permit the computation of linkage probability the same way we have done on our example table 3.3 to show $\ell-$diversity weaknesses. The third metric is the *Interval Disclosure* (ID) metric. Let v' being a masked value, the ID metric represents the size of the interval the original value of v' belongs (*i.e.* the smallest interval of values that are equal to v' after masking)

While those disclosure metrics may not fit generalization-based methods, other disclosure metrics have been proposed. In their paper, Louis-Philippe Sondeck et al. [80] proposed the *Combined Discrimination Rate* (CDR), an entropy-based metric. This metric aims to measure the capacity of a set of generalized attributes to *refine* the sensitive attribute (*i.e.* the refinement corresponds to how much a sensitive attribute is associated to a set of generalized attributes). They define it as follows:

$$CDR_X(Y) = 1 - \frac{H(X \mid Y)}{H(X)} \tag{3.20}$$

with X the *sensitive* attribute and Y a set of attributes. Note that $1 \leq |Y| \leq |\mathcal{A}|$ with $\mathcal{A}$ the set of attributes in $\mathcal{D}$. In our example table 3.3, the result of the CDR metric is illustrated by table 3.5.

In the table 3.5, an attribute A is denoted A^* if it is the generalized version of A, *e.g.* Age* is the generalization of the Age attribute and is represented by intervals. Let us remember that the table 3.3 is the $k-$anonymous version of table 3.2, so Age* (*resp.* Age) is corresponding to the attribute Age in table 3.3 (*resp.* table 3.2). To explain the result of $CDR_X(Y)$, let us take $X =$ Condition and $Y =$ Age*. Firstly, the *Cancer* condition is associated to four persons, the *Heart Disease* condition to two persons and the *Viral*

Infection condition to three persons, leading us to the following conclusion:

$$\begin{aligned} H(X) =& -\left[\frac{4}{9}\cdot\log\frac{4}{9}+\frac{2}{9}\cdot\log\frac{2}{9}+\frac{3}{9}\cdot\log\frac{3}{9}\right] \\ H(X) =& 1.53 \end{aligned} \tag{3.21}$$

On the other hand, there are two people with an age lesser or equal to 35 with the *Cancer* condition and one with the *Heart Disease* condition which give us the following result:

$$\begin{aligned} H(X|\text{Age}^* = [25,35]) =& -\left[\frac{2}{9}\cdot\log\frac{2}{3}+\frac{1}{9}\cdot\log\frac{1}{3}\right] \\ H(X|\text{Age}^* = [25,35]) =& 0.31 \end{aligned} \tag{3.22}$$

Combining $H(X|\text{Age}^* = [25,35])$ with $H(X|\text{Age}^* = [38,44])$ and $H(X|\text{Age}^* = [45,70])$, we obtain that $CDR_{\text{Condition}}(\text{Age}^*) = 0.25$. The lower CDR is the lower the attribute X can be deduce from the value of Y. The CDR metric can be used as a disclosure risk metric. It is important to note that using an identifier as Y value lead to $CDR_X(Y) = 1$, so in our example, $CDR_{\text{Condition}}(\text{ZIP}, \text{Age}) = 1$. Since the CDR metric does not take into account the potential proximity of different sensitive values, another metric was proposed from the same authors [81], the *Semantic Discrimination Rate* (SeDR). The $SeDR$ consider the fact that sensitive values may have a similar semantic meaning. For example, a throat cancer is more similar to a lung cancer than a flu. The idea of the SeDR metric is to compute the CDR metric over the *semantic* of sensitive values. For example, let a medical dataset made of the following sensitive values set: $S_1 = \{$throat cancer, lung cancer, breast cancer, Alzheimer, Parkinson, Huntington, flue, varicella$\}$. The SeDR metric could be computed on the following set: $S_2 = \{$cancer, neurodegenerative disease, other disease$\}$. The anonymised dataset would contain value from S_1 but the SeDR metric could be used by computing the CDR metric considering sensitive values as their semantic meaning in S_2.

Despite the important interest *disclosure metrics* give to produce a great tradeoff data utility/privacy-preserving, we consider the use of these metrics differing from the concept of $k-$anonymity. The use of these metrics to create an anonymised dataset would be another anonymity concept.

3.5 Conclusion

Many methods and different approaches have been proposed to protect individuals privacy. Partition-based methods and differential privacy mechanisms are two distinct approaches offering different characteristics. Many legislators such as the GDPR [75] are pushing user empowerment initiatives bringing new perspectives. Despite this interest, a small number of privacy-preserving methods have been proposed to give to individuals the opportunity to personalise their security. While partition-based approaches lack from the possibility to measure the data leak induced by the publication of anonymised datasets, the differential privacy approach lacks from the computations allowed on anonymised dataset, *i.e.* the differencial privacy aims to produce an anonymised dataset for a set of queries while partition-based approaches aim to produce a general anonymised dataset letting the possibility to answer any query. Also, to add user empowerment, it is necessary that users are able to undestand their options. Since fixing the differential privacy parameter is a cumbersome and not intuitive task, out of reach of lambda individuals, using partition-based approaches is a better solution to propose to users the personalisation of their protection. Note that some partition-based approaches can be seen as differential privacy mechanisms such as the stochastic $t-$closeness (an extension of $t-$closeness) as shown by J. Domingo-Ferrer and J. Soria-Comas [19].

Many partition-based approaches have been researched. However, most of them compute the same protection for all individuals. Many studies have proven that individuals do not share the same opinion about privacy. Some persons prefer a high preservation of their privacy while others accept a low protection. Partition-based approaches lacks from user empowerment, not letting individuals express their will. The $k-$anonymity approach is the most close to bring personalisation in practice. While algorithms computing optimal $k-$anonymity show no feasibility due to the fact that optimal $k-$anonymity is NP-Hard [72], others are heuristics which try to get close to optimal $k-$anonymity. While their significant number permits to adapt the computation of $k-$anonymity to the desired use case (*i.e.* need of more or less scalability and data quality), most of them propose a uniform computation of $k-$anonymity leading to a lack of individuals personalisation. The few amount of user empowerment propositions we can actually find in the state of the art of $k-$anonymity methods are not satisfactory. For in-

stance, some of them propose the modification of sensitive information which keeps them away from the primary objective of $k-$anonymity that is to say keeping sensitive information unspoiled while making it impossible to link it to a physical individual. Other user empowerment solutions propose to let users choosing the k they are willing. Although these approaches are interesting, they only deal with spatiotemporal data. Spatiotemporal data are the main interest of many studies. Indeed, with the development of location based services, the collect of spatiotemporal plays an important role to deduce knowledges such as points of interest (POIs) detection or supermarket traffic flow. Even if many anonymisation methods of spatiotemporal data exist to preserve individuals' privacy, other studies show the possibility to de-anonymise such anonymisation. For example, S. Gambs et al. [31] propose a way to de-anonymise these data using *Mobility Markov Chains* (MMCs) originally presented on an article from the same authors [32]. A MMC is associated to an individual and is composed of a set of POIs and a set of probabilities of moves from a POI i to another one j the individual can made. This set of probalities may be represented by a matrix. Their idea is to use the association of two metrics they defined and which aim to mesure the distance from two MMCs. They tried to use the MMC associated to an individual to find if he appears within a set of anonymous MMCs. Although MMCs are structures that must be trained, the set of anonymous MMCs has not necessary be trained with the same dataset of the MMC the attacker know. They showed that they could identify the user highlighting the weakness of spatiotemporal data anonymisation methods. Even if spatiotemporal data are the main interest of many studies, spatiotemporal data are far from covering the whole domain of data science. Despite this interest, no one to the best of our knowledge has studied the computation of $k-$anonymity with individual personalisation on common data attributes (*i.e.* categorical, ordinal, nominal, numerical, ...).

On the other hand, the way micro-data are managed today involves the use of a trusted third party. However, with the advent of cloud computing, outsourcing costly operations to a third party is common. While, in practice, computations over micro-data are often done under a confidentiality clause, data leakages happen pretty often. Some solutions propose to outsource computations to an untrusted third party while ensuring this third party cannot access to any sensitive data using different methods such as homomorphic encryption systems or secure cryptoprocessor. A major drawback from these approaches are their lack of scalability. Other approaches try to restrict data

computations to the least third parties. While some use a complex system to ensure responses to queries do not leak personal data, another uses a large amount of trusted devices sharing smart-card characteristics coupled to a performent untrusted server infrastucture to compute costly operations without leaking any personal information to the untrusted third party. However, this approach lacks from guarantees of privacy-preservation on answers to queries it gives.

The previous approaches try to fit *Big Data* use cases. In spite of the popularity of the *Big Data* concept, there is some use cases which do not fall under the *Big Data* domain. In those cases, small datasets are much more common and cannot be anonymised the same way as *Big Data* datasets. Indeed these small datasets turn optimal $k-$anonymity computations feasible. Even so, optimal $k-$anonymity methods do not propose any personalisation of $k-$anonymity.

This manuscript focuses on three major objectives:

- Proposing a way to securely compute queries while ensuring that user privacy constraints are satisfied during the process.
- Offering the possibility to empower users when a trusted institute aims to publish a $k-$anonymous dataset.
- Suggest a method to compute optimal $k-$anonymity under users security personalisations.

Chapter 4

Personalised $k-$Anonymity Through SQL

In this chapter we present an approach in which users specify their individual anonymity constraints. In the other hand, a querier declares privacy guarantees through SQL query when willing to achieve a statistics study. This permits to users to automatically determine their participation regarding those guarantees. In this approach, the querier can also define data generalisation processes giving higher guarantees to fit more users constraints. Our approach is built upon an asymmetric architecture based on trusted hardware and allows SQL computations without any privacy leak.

4.1 Introduction

The way microdata is anonymised and processed today is far from being satisfactory. Let us consider how a national statistical study is managed, *e.g.* computing the average salary per geographic region. Such a study is usually divided into 3 phases: (1) the statistical institute (assumed to be a *trusted third party*) broadcasts a query to collect raw microdata along with *anonymity guarantees* (*i.e.*, a privacy parameter like k in the case of $k-$*anonymity* or ϵ in the case of *differential privacy*) to all users ; (2) each user consenting to participate transmits her microdata to the institute ; (3) the institute computes the aggregate query, while respecting the announced anonymity constraint.

This approach has two important drawbacks:

1. The anonymity guarantee is defined by the querier (*i.e.* the statistical institute), and applied uniformly to all participants. If the querier decides to provide little privacy protection (*e.g.* a small k in the $k-$*anonymity* model), it is likely that many users will not want to participate in the query. On the contrary, if the querier decides to provide a high level of privacy protection, many users will be willing to participate, but the quality of the results will drop. Indeed, higher privacy protection is always obtained to the detriment of the quality and thus utility of the sanitized data.

2. The querier is assumed to be trusted. Although this could be a realistic assumption in the case of a national statistics institutes, this means it is impossible to outsource the computation of the query. Moreover, microdata centralization exacerbates the risk of privacy leakage due to piracy (Yahoo and Apple recent hack attacks are emblematic of the weakness of cyber defenses[1]), scrutinization and opaque business practices. This erodes individuals trust in central servers, thereby reducing the proportion of citizen consenting to participate in such studies, some of them unfortunately of great societal interest.

The objective of this chapter is to tackle these two issues by reestablishing *user empowerment*, a principle called by all recent legislations protecting

[1]Yahoo 'state' hackers stole data from 500 million users - BBC News. https://www.bbc.co.uk/news/world-us-canada-37447016

the management of personal data [75]. Roughly speaking, user empowerment means that the individual must keep the control of her data and of its disclosure in any situation. More precisely, this chapter makes the following contributions:

- propose a query paradigm incorporating personalized privacy guarantees, so that each user can trade her participation in the query for privacy protection matching her personal perception of the risk,
- present and discuss possible semantics of personalized privacy queries,
- propose efficient algorithms to manage the collection and distributed computation of queries over large quantities of data
- provide a secure decentralized computing framework guaranteeing that the individual keeps her data in her hands and that the query issuer never gets cleartext raw microdata and sees only a sanitized aggregated query result matching all personalized privacy guarantees,
- conduct a performance evaluation and large scale implementation on a real dataset demonstrating the effectiveness and scalability of the approach.

The rest of the chapter is organized as follows. Section 4.2 presents related works and background materials allowing to precisely state the problem addressed. Section 4.3 details how to manage personalized queries in SQL. Section 4.4 covers a discussion of the semantics of this model. Section 4.5 discusses the complex algorithmic issues that arise due to the use of personalized anonymity. Section 4.6 shows the results of our implementation, demonstrating the efficiency of the approach in real case scenarios. Finally, Section 4.7 concludes.

4.2 Background Informations

4.2.1 Reference computing architecture

Concurrently with smart disclosure initiatives, the *Personal Information Management System* (PIMS) paradigm has been conceptualized [1], and is emerging in the commercial sphere (*e.g.* Cozy Cloud, OwnCloud, SeaFile).

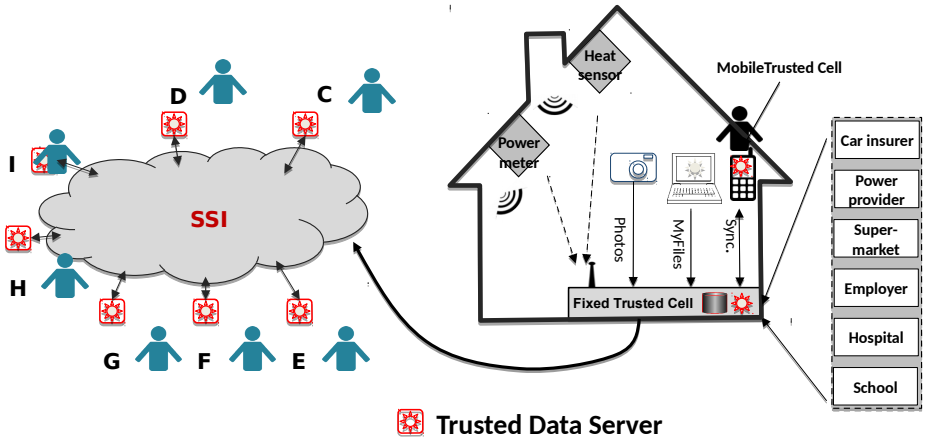


Figure 4.1: Trusted Cells reference architecture [4].

PIMS holds the promise of a Privacy-by-Design storage and computing platform where each individual can gather her complete digital environment in one place and share it with applications and other users, while preserving her control over her data. The *Trusted Cells architecture* presented in [4], and pictured in Figure 4.1, precisely answers the PIMS requirements by preventing data leaks during computations on personal data. Hence, we consider Trusted Cells as a reference computing architecture in this chapter.

Trusted Cells is a decentralized architecture by nature managing computations on microdata through the collaboration of two parties. The first party is a (potentially large) set of personal *Trusted Data Servers* (TDSs) allowing each individual to manage her data with tangible elements of trust. Indeed, TDSs incorporate tamper resistant hardware (*e.g.* smartcard, secure chip, secure USB token) securing the data and code against attackers and users' misusages. Despite the diversity of existing tamper-resistant devices, a TDS can be abstracted by (1) a Trusted Execution Environment and (2) a (potentially untrusted but cryptographically protected) mass storage area where the personal data resides. The important assumption is that the TDS code is executed by the secure device hosting it and thus cannot be tampered, even by the TDS holder herself.

By construction, secure hardware exhibit limited storage and computing resources and TDSs inherit these restrictions. Moreover, they are not necessarily always connected since their owners can disconnect them at will. A second party, called hereafter *Supporting Server Infrastructure* (SSI), is thus required to manage the communications between TDSs, run the distributed

query protocol and store the intermediate results produced by this protocol. Because SSI is implemented on regular server(s), *e.g.* in the Cloud, it exhibits the same low level of trustworthiness.

The resulting computing architecture is said *asymmetric* in the sense that it is composed of a very large number of low power, weakly connected but highly secure TDSs and of a powerful, highly available but untrusted SSI.

4.2.2 Reference query processing protocol

By avoiding delegating the storage of personal data to untrusted cloud providers, Trusted Cells is key to achieve user empowerment. Each individual keeps her data in her hands and can control its disclosure. However, the decentralized nature of the Trusted Cells architecture must not hinder global computations and queries, impeding the development of services of great interest for the community.

SQL/AA (SQL on Asymmetric Architecture) is a protocol to execute standard SQL queries on the Trusted Cells architecture [86, 85]. It has been precisely designed to tackle this issue, that is executing global queries on a set of TDSs without recentralizing microdata and without leaking any information.

Illustrated by figure 4.2, the protocol works as follows. Once an SQL query is issued by a querier (*e.g.* a statistic institute), it is computed in three phases: first the *collection phase* where the querier broadcasts the query to all TDSs, TDSs decide to participate or not in the computation (they send dummy tuples in that case to hide their denial of participation), evaluate the `WHERE` clause and each TDS returns its own encrypted data to the SSI. Second, the *aggregation phase*, where SSI forms partitions of encrypted tuples, sends them back to TDSs and each TDS participating in this phase decrypts the input partition, removes dummy tuples and computes the aggregation function (*e.g.* `AVG`, `COUNT`). Finally the *filtering phase*, where TDSs produce the final result by filtering out the `HAVING` clause and send the result to the querier. Note that the TDSs participating in each phase can be different. Indeed, TDSs contributing to the collection phase act as data producers while TDSs participating to the aggregation and filtering phases act as trusted computing nodes. The tamper resistance of TDSs is key in this protocol since a given TDS belonging to individual i_1 is likely to decrypt and aggregate tuples issued by TDSs of other individuals $i_2, \ldots, i_n$. Finally, note that the aggregation phase is recursive and runs until all tuples belonging to

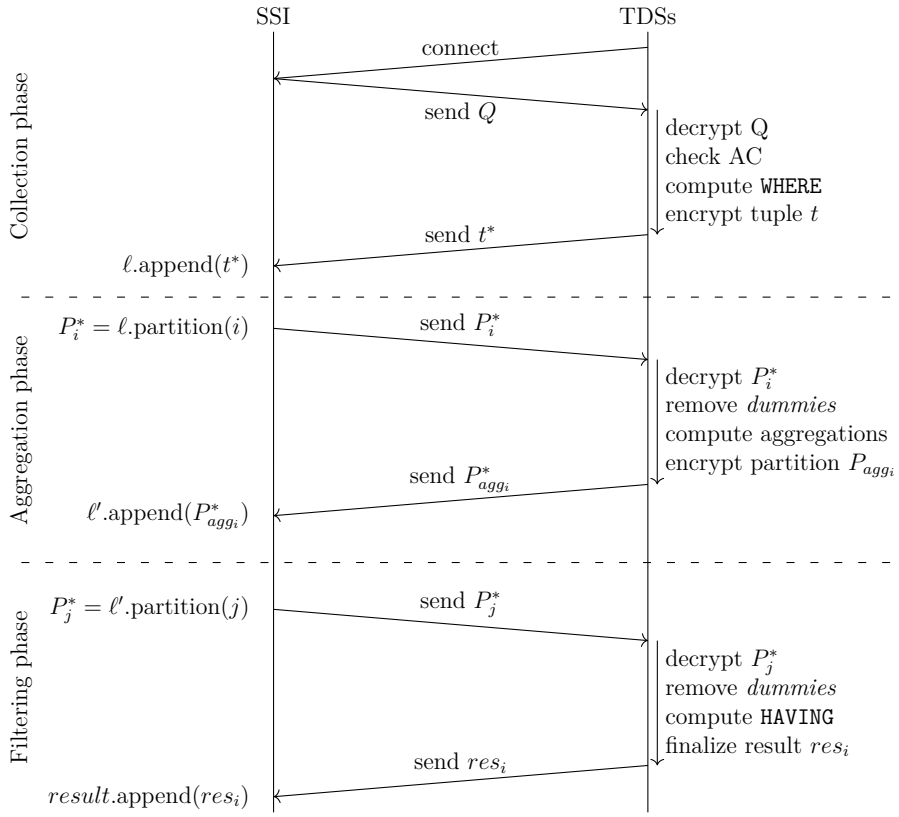


Figure 4.2: *SQL/AA* protocol [86, 85]

a same group have been actually aggregated. We refer the interested reader to [86, 85, 66] for a more detailed presentation of the SQL/AA protocol.

4.2.3 Problem Statement

In order to protect the privacy of users, queries must respect a certain degree of anonymity. Our primary objective is to push personalized privacy guarantees in the processing of regular statistical queries so that individuals can disclose different amount of information (*i.e.* data at different levels of accuracy) depending on their own perception of the risk. For the sake of simplicity, we consider SQL as the reference language to express statistical/aggregate queries because of its widespread usage. Similarly, we consider personalized privacy guarantees derived from the k-anonymity and ℓ-diversity models because (1) they are the most used in practice, (2) they are recommended by the European Union [25] and (3) they can be easily understood by individuals[2].

Hence, the problem addressed in this chapter is to propose a (SQL) query paradigm incorporating personalized (k-anonymity and ℓ-diversity) privacy guarantees and enforcing these individual guarantees all along query processing without any possible leakage.

4.3 Personalised Anonymity Guarantees in SQL

4.3.1 Modeling anonymisation using SQL

We make the assumption that each individual owns a local database hosted in her personal TDS and that these local databases conform to a common schema which can be easily queried in SQL. For example, power meter data (resp., GPS traces, healthcare records, etc) can be stored in one (or several) table(s) whose schema is defined by the national distribution company (resp., an insurance company consortium, the Ministry of Health, etc). Based on this assumption, the querier (i.e., the statistical institute) can issue regular SQL queries as shown by figure 4.3.

[2]The EU Article 29 Working Group mention these characteristics as strong incentives to make these models effectively used in practice or tested by several european countries (*e.g.* the Netherlands and French statistical institutes).

```
SELECT <Aggregate function(s)>
FROM <Table(s)>
WHERE <condition(s)>
GROUP BY <grouping attribute(s)>
HAVING <grouping condition(s)>
```

Figure 4.3: Regular SQL query form

For the sake of simplicity, we do not consider joins between data stored in different TDSs but internal joins which can be executed locally by each TDS are supported. We refer to [85] for a deeper discussion on this aspect which is not central to our work in this thesis.

Anonymity Guarantees are defined by the querier, and correspond to the k and ℓ values that will be achieved by the end of the process, for each group produced. They correspond to the commitment of the querier towards any query participant. Different k and ℓ values can be associated to different granularities of grouping. In the example pictured in figure 4.4, the querier commits to provide $k \geq 5$ and $\ell \geq 3$ at a (`City`,`Street`) grouping granularity and $k \geq 10$ and $\ell \geq 3$ at a (`City`) grouping granularity.

Anonymity Constraints are defined by the users, and correspond to the values they are willing to accept in order to participate in the query. Back to the example of figure 4.4, Alice's privacy policy stipulates a minimal anonymisation of $k \geq 5$ and $\ell \geq 3$ when `Consu` attribute is queried.

According to the anonymity guarantees and constraints, the query computing protocol is as follows. The querier broadcasts to all potential participants the query to be computed along with metadata encoding the associated anonymity guarantees. The TDS of each participant compares this guarantee with the individual's anonymity constraints. This principle shares some similarities with *P3P*[3] with the matching between anonymity guarantees and constraints securely performed by the TDS. If the guarantees exceed the individual's constraints, the TDS participates to the query by providing real data at the finest grouping granularity. Otherwise, if the TDS finds a grouping granularity with anonymity guarantees matching her constraints, it will participate, but by providing a degraded version of the data, to that coarser level of granularity (looking at figure 4.4, answering the `group by city, street` clause is not acceptable for Bob, but answering just with `city` is). Finally,

[3]https://www.w3.org/P3P/

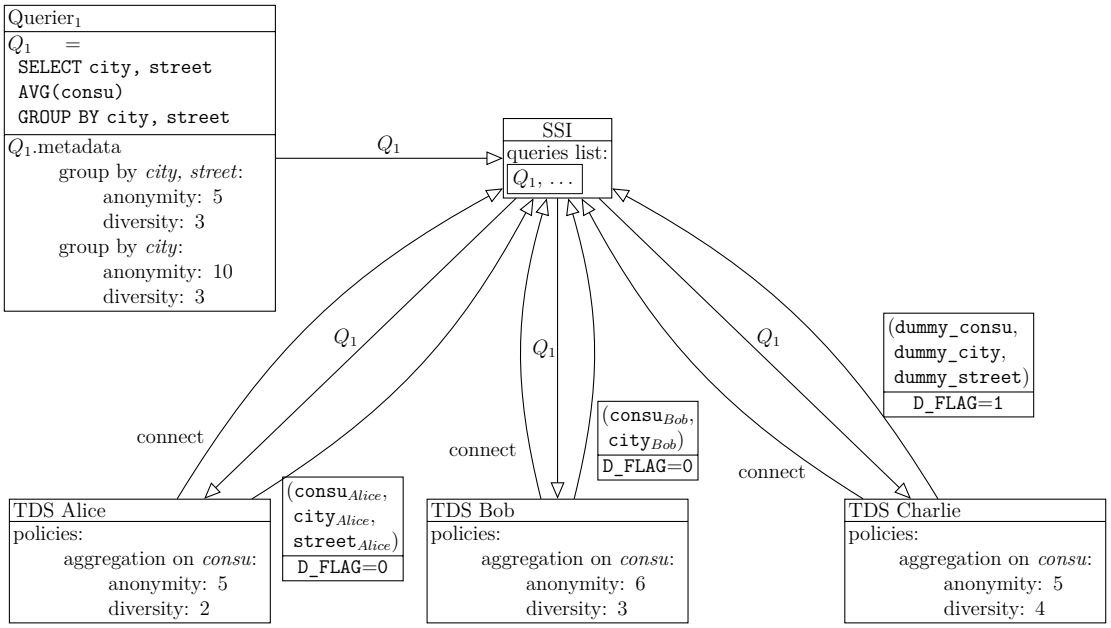


Figure 4.4: Example of collection phase with anonymity constraints

if no match can be found, the TDS produces fake data (called *dummy tuples* in the protocol) to hide its denial of participation. Fake data is required to avoid the querier from inferring information about the individual's privacy policy or about her membership to the `WHERE` clause of the query.

Figure 4.4 illustrates this behavior. By comparing the querier anonymity guarantees with their respective constraints, the TDSs of Alice, Bob and Charlie respectively participate with fine granularity values (Alice), with coarse granularity values (Bob), with dummy tuples (Charlie).

The *ODRL*[4] working group is looking at some issues similar to the expression of privacy policies. However, this paper does not discuss about how users can express their privacy policy in a standard way.

4.3.2 The ^{k_i}SQL/AA protocol

We now describe our new protocol, that we call ^{k_i}SQL/AA to show that it takes into account many different k values of the i different individuals. ^{k_i}SQL/AA is an extension of the SQL/AA protocol [86, 85] where the en-

[4]https://www.w3.org/community/odrl/

city	street	AVG(salary)	COUNT(*)	COUNT (DISTINCT salary)
Le Chesnay	Dom. Voluceau	1500	6	4
Le Chesnay	******	1700	9	6
Bourges	Bv. Lahitolle	1600	3	3
Bourges	******	1400	11	7

(a) Example of a post aggregation phase result

Attributes	k	ℓ
city, street	5	3
city	10	3

(b) Privacy guarantees of the query

city	street	AVG(salary)
Le Chesnay	Dom. Voluceau	1500
Bourges	******	1442.86

(c) Data sent to the querier

Table 4.1: Filtering phase

forcement of the anonymity guarantees have been pushed to the *collection*, *aggregation* and *filtering* phases.

Collection phase: After TDSs download the query, they compare the anonymity guarantees announced by the querier with their own anonymity constraints. As discussed above (see Section 4.3.1) TDSs send real data at the finest grouping granularity compliant with their anonymity constraints or send a dummy tuple if no anonymity constraint can be satisfied.

Aggregation phase: To ensure that the anonymisation guarantees can be verified at the filtering phase, clauses `COUNT(*)` and `COUNT(DISTINCT A)` are computed in addition to the aggregation asked by the querier. `COUNT(*)` will be used to check that the $k-$anonymity guarantee is met while `COUNT(DISTINCT A)`[5] are computed in addition to the aggregation asked by the querier. As explained next, these clauses are used respectively to check the $k-$anonymity and $\ell-$diversity guarantees). If tuples with varying grouping granularity enter this phase, they are aggregated separately, *i.e.* one group per grouping granularity.

Filtering phase: Besides `HAVING` predicates which can be formulated by the querier, the `HAVING` clause is used to check the anonymity guarantees.

[5]Since this clause is an holistic function, we can compute it while the aggregation phase by adding naively each distinct value under a list or using a cardinality estimation algorithm such as *HyperLogLog* [26].

Typically, $k-$anonymity sums up to check `COUNT(*)`$\geq k$ while $\ell-$diversity is checked by `COUNT(DISTINCT A)`$\geq \ell$. If these guarantees are not met for some tuples, they are not immediately discarded. Instead, the protocol tries to merge them with a group of coarser granularity containing them. Let us consider the example of Table 4.1a. The tuple `(Bourges, Bv.Lahitolle, 1600)` is merged with the tuple `(Bourges, ******, 1400)` to form the tuple `(Bourges, ******, 1442.86)`. Merges stop when all guarantees are met. If, despite merges, the guarantees cannot be met, the corresponding tuples are removed form the result. Hence, the querier will receive every piece of data which satisfies the guarantees, and only these ones, as shown on Table 4.1c. In this same example, note that another choice could have been done regarding tuple `(Le Chesnay, ******)`. Instead of removing it because it does not meet the privacy constraints, it could have been merged with tuple `(Le Chesnay, Dom. Voluceau)`. The result would lose in precision but will gain in completeness. We discuss more deeply the various semantics which can be associated to personalized anonymisation in Section 4.4.

How to generalize. The example pictured in Table 4.1 illustrates data generalization in a simplistic case, that is a Group By query performed on a single attribute which can be expressed with only two levels of precision `<City,Street>` and `<City>`. In the general case, Group By queries can be performed on several attributes, each having multiple levels of precision. To reach the same k and ℓ values on the groups, the grouping attributes can be generalized in different orders, impacting the quality of the result for the querier. For instance, if the `GroupBy` clause involves two attributes Address and Age, would it be better to generalize the tuples on Address (*e.g.* replacing `<City,Street>` by `<City>`) or on Age (replacing exact values by intervals) ? The querier must indicate to the TDSs how to generalize the data to best meet her objectives, by indicating which attributes to generalize, in which order, and what privacy guarantees will be enforced after each generalization. In the following example, we consider the UCI Adult dataset [59], we define a `GroupBy` query `GB` on attributes `Age`, `Workclass`, `Education`, `Marital_status`, `Occupation`, `Race`, `Gender`, `Native_Country` and we compute the average `fnlwgt` operation `OP=AVG(fnlwgt)`. `MD` represents the metadata attached to the query. Each metadata indicates which k and ℓ can be guaranteed after a given generalization operation. Depending on the attribute type, generalizing an attribute may correspond to climbing up in a generalization hierarchy (for categorical attributes such as `Workclass` or `Race`) or replacing a value by an interval of greater width (for numeric val-

ues such as `Age` or `Education`). The `del` operation means that the attribute is simply removed. The ordering of the metadata in `MD` translates the querier requirements.

```
GB= Age, Workclass, Education, Marital_status,
    Ocupation, Race, Gender, Native_Country;
OP= AVG(fnlwgt);
MD= :                    k=5 l=3,
    age->20:             k=6 l=3,
    workclass->up:       k=8 l=4,
    education->5:        k=9 l=4,
    marital_status->up: k=9 l=4,
    occupation->up:      k=10 l=4,
    race->up:            k=11 l=5,
    gender->del:         k=14 l=6,
    native country->del:k=15 l=7,
    age->40:             k=17 l=8;
```

Figure 4.5: ^{k_i}SQL/AA query example

4.3.3 Differences in importance of attributes

The way data is anonymised is in general defined by the algorithm applying generalizations. Most of the time all attributes are considered as equivalent in term of importance. The problem is that generalizing an attribute can have a higher impact than generalizing another. Let us represent the degradation of an attribute by a percentage. For instance, generalizing sex data from *Female/Male* to *individual* leads to a 100% information loss on that attribute. In contrary, generalizing the age by making intervals of ages two by two, generates low information loss. This difference of generalization granularities can lead many anonymisation methods to generate high information loss. Another problem is the difference of minimum utility statisticians need. For example, a statistician could feel that utility falls to nill when an attribute is generalized too much, even if it is not totally anonymised. There is a gap where the utility falls to nill and this gap depends on the objective of the query. Most anonymisation methods aim to optimize a metric (see section 3.4) which is not always the same objective as the querier. That is why, supervised methods such as the $\mu-argus$ framework from the CASC project [42] or the *R* package *sdcMicro* [84] are often preferred by

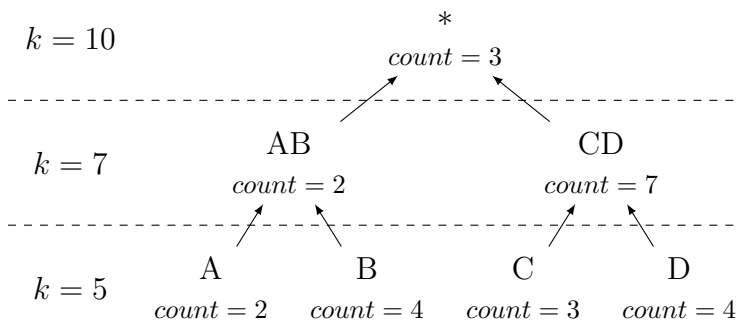


Figure 4.6: Post-aggregation phase example

statisticians. Our approach also follows the idea that a statistician or a querier should be able to specify how the generalization process occurs. The ^{k_i}SQL/AA protocol is a *semi-supervised* approach where the querier can specify the different steps of the anonymisation process while not having a view of the data. By doing so, the querier is able to define the granularity of each attribute generalization he is willing to have.

4.4 Semantic Issues Linked to K_i-Anonymisation

The example presented in Table 4.1 highlighted the fact that no single personalized anonymisation semantics fits all situations. Typically, personalized anonymisation introduces an interesting tradeoff between preciseness and completeness of the query results. Figure 4.6 illustrates this tradeoff and shows that two different result tables might both respect the privacy constraints, while containing different tuples. In other words, it is important to be able to define the formal characteristics that can be guaranteed, even though some are contradictory. We call *semantics* of a k_i-anonymised query the rules that lead to publishing tuples under privacy guarantees. These semantics are evaluated during the filtering phase of the query.

Let Q be a query computing an aggregation on a set of tuples, grouping on an attribute a of finite domain $dom(a)$. We note $|dom(a)|$ the size of the domain. For the sake of simplicity, we assume that the querier specifies the generalization process by means of a labeled generalization *tree* (thus each value has only one parent) which explains how each value must be generalized, and which indicates the privacy guarantees enforced at each generalization

level of a.

We plot on Figure 4.6 an example where $dom(a) = \{A, B, C, D\}$. The example represents the post-aggregation result of query Q (before the filtering phase). For readability purposes, the k privacy guarantee labels are indicated on the left. Each node shows the total number of participants in the given aggregation group. It is important to note that at this point, a microdata tuple will only have contributed to a *single* group, depending on its privacy preferences.

Example: Group AB is composed of the tuples of 2 participants, with values A or B for a, and with a privacy guarantee of $k \geq 7$. Clearly as only 2 participants contributed to the group in this case, this guarantee cannot be enforced, and thus this group cannot be published *as* is in the resulting query answer.

Different decisions, which will impact the semantics of the Group By query, can be made regarding this group: (i) discard the group, and the corresponding microdata tuples, (ii) merge it with a more general group (*i.e.* a parent node in the generalization tree), (iii) add tuples from a more specific but compatible group to it. We discuss next different realistic semantics for the Group By query and how to achieve them. Two concurrent objectives can actually be pursued: generalizing the microdata tuples the least possible or dropping the least possible microdata tuples. Note that when publishing the results of such queries, although a microdata tuple is only counted once, the domains of the groups can overlap.

1. ***Selective* semantics.** These semantics prioritize the precision of the result over its completeness, while enforcing all privacy guarantees. In other words, the querier wants to keep the best quality groups as possible, that is at the finest generalization level, and accepts to drop groups that do not achieve the privacy constraints.

2. ***Complete* semantics.** Conversely, these semantics prioritize the completeness of the result over its precision. The querier wants to keep the most tuples as possible, and accepts to generalize and merge groups to achieve this objective.

Selective semantics are rather straightforward since they correspond to a locally optimizable algorithm. They can be implemented as the algorithm 1.

With this algorithm 1, the microdata are generalized along the generalization tree until the group cardinality reaches the privacy guarantee of a

Algorithm 1 *Selective* algorithm

```
procedure selective(List L)
    while N = L.pop() do
        if satisfyPrivacy(N) then
            publish(N)
        else
            if hasParent(N) then
                N.mergeTo(parent(N))
            end if
        end if
    end while
end procedure
```

given node and are then published with this level of precision. Once a node content is published, it cannot be used by a higher level node to help it get published. Thus a node N cannot be discarded if a higher level node N' is published.

Example: Groups A and B do not respect the privacy constraint, they are thus merged with Group AB. The same holds for groups C and D which are merged with group CD. Groups AB and CD (including the newly generalized tuples) are both published. Group $*$ does not respect the privacy constraint and is discarded.

Complete semantics are more difficult to achieve. Indeed, the objective here is to globally optimize the number of tuples published, and only then look at the quality of the generalization. The problem can be formalized as a constrained optimization problem with two (ordered) objective functions. F_1 is the priority optimization function which counts the number of microdata tuples published. F_2 is the secondary optimization function which evaluates the overall quality of groups. Functions F_2 could be defined to take into account the semantic relatedness among nodes of the generalization tree by using appropriate metrics (*e.g.* Wu and Palmer[94] metric), but for the sake of simplicity we chose to define the overall quality as the $\sigma_g \in \{groups\}(depth(g) \cdot count(g))$. Maximal quality is obtained if all the microdata tuples are published at maximal depth. Dropped tuples induce maximal penalty in quality, since they do not contribute to the score. Also note that it is not possible to generalize only a subset of the microdata tuples composing one of the initial groups. A group must be

generalized completely or not at all. However, descendant groups could be generalized more than some of their ancestor groups. For instance, group CD might be published as is since it already satisfies the privacy guarantee, and groups C and D might be generalized to $*$ to form a group of cardinality 10 which can also be published (instead of dropping the three initial elements in $*$). Hence, regarding F_1, various combinations are valid like $\{(*, count = 10), (AB, count = 8), (CD, count = 7)\}$, $\{(*, count = 18), (CD, count = 7)\}$ or $\{(*, count = 11), (CD, count = 14)\}$. In this example, F_2 should lead to select the former one.

The practical difficulty comes precisely from the computation of this secondary optimization function F_2 (not to mention having to chose what function to use). Indeed, otherwise a trivial answer would simply generalize all the microdata tuples to $*$, and publish a single group. In a centralized context, a (non linear) constraint solver could be used to compute the optimal result to this problem. However, in a distributed context, generalization must take place during the distributed filtering phase. Thus we propose to use a greedy heuristic to simplify decision making in the *complete* semantics approach.

Algorithm 2 Greedy *complete* algorithm.

```
procedure complete(Ordered List (lower node first) L)
    R ←empty list
    while N = L.pop() do
        if satisfyPrivacy(N) then
            R.push(N)
        else if hasDescendant(N) then
            P ← descendant(N)
            R.remove(P)
            P.mergeTo(N)
        else if hasParent(N) then
            N.mergeTo(parent(N))
        end if
        publish(R)
    end while
end procedure
```

The difference between these two algorithms is that the second algorithm accepts to generalize a node which could be published as is, if it can help a

higher node to get published. Also note that another semantic choice that must be made when implementing the algorithm is when deciding which descendant node to merge, when several possibilities exist.

Example: The algorithm behaves the same as the *selective* algorithm until it has groups AB and CD in R. Then it must decide whether to merge AB or CD with $*$. The decision can be made by chosing the smallest group that still respects the privacy constraint. In this case, as $|AB| = 8$ and $|CD| = 14$, we would chose to merge AB with $*$ and produce $|*| = 11$ and $|CD| = 14$. However we see that the optimal solution would have been to merge C and D with $*$ or merge CD with $*$ and generalize C and D to CD, thus producing $|AB| = 8$, $|CD| = 7$ and $|*| = 10$.

Conclusion: Several different semantics can be applied in order to decide how to publish the groups while respecting privacy constraints. There is no ideal choice in absolute terms and the decision should be application driven. However, besides the preciseness and completeness of the obtained result, the cost of implementing these semantics may widely differ. While implementing the *selective* semantics is straightforward, optimizing the computation of the *complete* semantics can be rather complex and costly, especially in a decentralized context.

4.5 Algorithmic Issues Linked to K_i-Anonymisation

The implementation of ^{k_i}SQL/AA builds upon the *secure aggregation* protocol of SQL/AA [86, 85], recalled in Section 4.2.2. Secure aggregation is based on non-deterministic encryption as AES CBC to encrypt tuples. Each TDS encrypts its data with the same secret key but with a different initialization vector such that two tuples with the same value have two different encrypted values. Hence all TDSs, regardless of whether they contributed to the collection phase or not, can participate in a distributed computation and decrypt all intermediate results produced by this computation, without revealing any information to the SSI. Indeed, the SSI cannot infer personal information by the distribution of non deterministically encrypted values. Injecting personnalization in this protocol has no impact on the cryptographic protection. However, this impacts the internal processing of each phase of the protocol, as detailed next.

4.5.1 Collection phase

The confrontation between the querier's privacy guarantees and the TDS's privacy constraints, along with the potential data generalization resulting from this confrontation, are included in the collection phase. The resulting algorithm is given below (see Algorithm 3). As in most works related to data anonymisation, we make the simplifying assumption that each individual is represented by a single tuple. Hence, the algorithm always returns a single tuple. This tuple is a dummy if the privacy constraints or the `WHERE` clause cannot be satisfied.

Algorithm 3 Collection Phase.

```
procedure COLLECTION_PHASE(Query Q)
    t ← getTuple(Q)
    p ← getConstraints(Q)
    g ← getGuarantees(Q)
    i ← 0
    if verifyWhere(t, Q) then
        while g_i < p do
            if not canGeneralize(t) then
                t ← makeDummy(Q)
            else
                t ← nextGeneralization(t)
                i ← i + 1
            end if
        end while
    else
        t ← makeDummy(Q)
    end if
    return Encrypt(t)
end procedure
```

4.5.2 Aggregation phase

To make sure that aggregations are entirely computed, the SSI uses a divide and conquer partitioning to make the TDS compute partial aggregations on

partitions of data. The aggregation phase, as implemented in SQL/AA [86, 85], is illustrated by Figure 4.7.

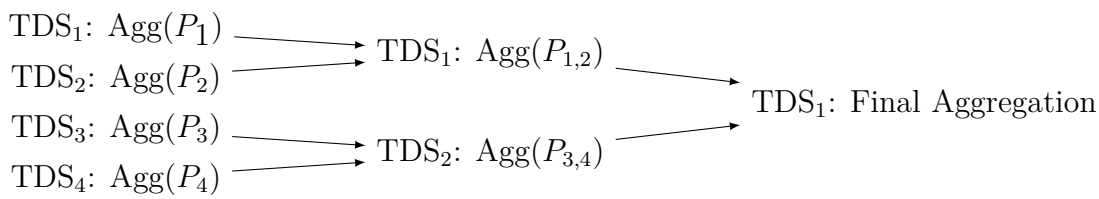


Figure 4.7: Aggregation phase with four partitions.

The weakness of this way of computing the aggregation is the work load put on the unique TDS performing the final aggregation. The limited amount of TDS RAM reduces the number of different groups (*i.e.* given by the `GroupBy` clause) which can be managed by a single TDS using RAM only. Moreover, the larger the partitions to merge, the larger the time spent by that TDS to do the computation and the greater the burden put on the TDS owner who cannot easily perform other tasks in parallel. This limitation is magnified in our context since personalized anonymisation may lead to a larger number of groups since initial groups may appear at different granularity levels.

We propose two algorithms overcoming this limitation. The first algorithm, called the *swap-merge* algorithm, is a direct adaptation of the SQL/AA aggregation algorithm where partitions bigger than the TDS RAM are simply swapped in NAND Flash. When it comes to compute aggregations on two sorted partitions saved on NAND Flash, the TDS will use a simple merge sort to build the result partition. The corresponding algorithm is straightforward and not detailed further due to space limitation. The benefit of this adaptation is to accomodate an unlimited amount of groups while keeping logarithmic the number of aggregation phases. However, this benefit comes at the price of ever increasing the load imposed on TDSs at the root of the aggregation tree due to NAND Flash I/Os. Note that executing the initial SQL/AA aggregation algorithm in streaming to avoid resorting to NAND swapping on each TDS would reveal the tuple ordering to the SSI. This option is thus discarded.

The second algorithm, illustrated by figure 4.8, does not rely on NAND Flash swapping and is called the *network-merge*. The idea is to use a bubble

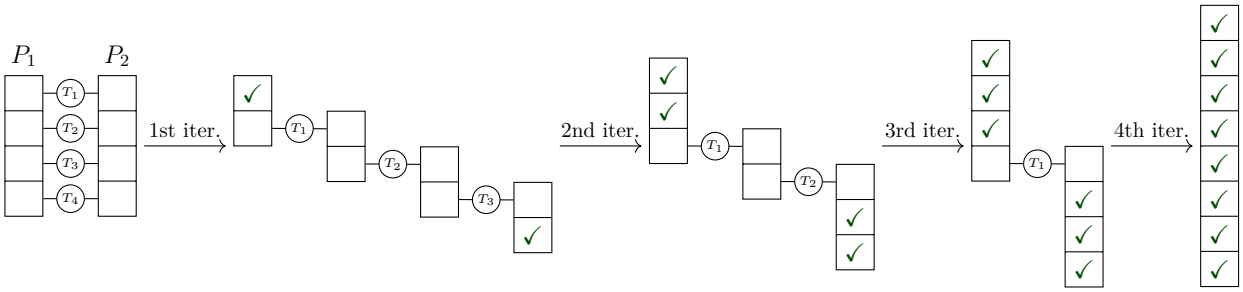


Figure 4.8: Merging sorted partitions without NAND access

sort like algorithm to merge two sorted partitions.

The figure 4.8 illustrates the merge of two sorted partitions P_1 and P_2. We assume that each partition is in turn decomposed into n frames ($n = 4$ in the figure) in order to accomodate the RAM capacity of a TDS. More precisely, the maximal size of a frame is set to half of the TDS RAM size (so that two frames of two different partitions can fit in RAM and be merged without swapping), minus a RAM buffer required to produce the merge result. Let P_i^j denote the jth frame of partition P_i and let P_r denote the partition resulting from the merge of P_1 and P_2. Assuming that P_1 and P_2 are internally sorted in ascending order on the grouping attributes, the merge works as follows. For each $j = 1..n$, the jth frames of P_1 and P_2 are pairwise sent to any merger TDS[6] which produces in return the jth and $(j+1)^{th}$ sorted frames of P_r. All elements in P_r^j are smaller or equal to the elements of P_r^{j+1}. By construction, after the first iteration of the protocol, frame P_r^1 contains the smallest elements of P_r and frame P_r^n the greatest ones. Hence, these two frames do not need to participate to the next iterations. Hence, after k iterations, $2 \cdot k$ frames of P_r are totally sorted, the smallest elements being stored in frames $P_r^{1..k}$ and the greatest ones being stored in frames $P_r^{n-k..n}$. The algorithm, presented in 4 is guaranteed to converge in a linear number of iterations. For readability concern and without loss of generality, we suppose that P_1 and P_2 have the same size.

Hence, algorithm *network-merge* can accomodate any number of groups without resorting to NAND Flash swapping. The price to pay is a linear (instead of logarithmic for *swap-merge* algorithm) number of iterations. The negative impact on the latency of the global protocol is low compared to the

[6]each TDS can contribute to any phase of the protocol, depending on its availability, independently of the fact that it participated to the collection phase.

Algorithm 4 Partition Merger Algorithm

```
procedure NETWORK-MERGE(Partition P_1, Partition P_2, Command cmd)
    Let top, bot, mid, q_1, q_2 empty partitions
    for i from 1 to NumberOfFrame(P_1) do
        Send(P_1^i, P_2^i, cmd) to TDS_i
    end for
    for i from 1 to NumberOfFrame(P_1) do
        P_r ← receive() from TDS_i
        if i == 1 then
            top.append(P_r^1)
        else
            q_1.append(P_r^1)
        end if
        if i == NumberOfFrame(P_1) then
            bot.append(P_r^2)
        else
            q_2.append(P_r^2)
        end if
    end for
    if NotEmpty(q_1) then
        mid ← NETWORK-MERGE(q_1, q_2, cmd)
    end if
    return top + mid + bot
end procedure
```

latency of the collection phase itself. But the benefit is high in terms of TDS availability. Thanks to this algorithm, each participating TDS contributes to a very small part of the global computation and this part can additionally be bounded by selecting the appropriate frame size. This property may be decisive in situations where TDSs are seldom connected.

4.5.3 Filtering Phase

The filtering phase algorithm is given by Algorithm 5.

Algorithm 5 Filtering Phase.

```
procedure FILTERING_PHASE(Query Q, TuplesSet T)
    sortByGeneralizationLevel(T)
    g ← getGuarantees(Q)
    for i from 0 to MaxGeneralizationLevel(Q) do
        for t ∈ T_i do
            t ← decrypt(t)
            if verifyHaving(t, Q) then
                result.addTuple(t)
            else if canGeneralize(t) then
                t ← nextGeneralization(t)
                T_{i+1}.addTuple(t)
            end if
        end for
    end for
    return result
end procedure
```

First, the algorithm sorts tuples of the aggregation phase by generalization level, making multiple sets of tuples of same generalization level. Function `verifyHaving` checks if `COUNT(*)` and `COUNT(Disctinct)` match the anonymisation guarantees expected at this generalization level. If so, the tuple is added to the result. Otherwise, it is further generalized and merged with the set of higher generalization level. At the end, every tuple which cannot reach the adequate privacy constraints, despite achieving maximum generalization, is not included in the result.

4.6 Experimental Evaluation

We have implemented the ^{k_i}SQL/AA protocol on equivalent open-source hardware used in [86, 85]. The goal of this experimental section is to show that there is very little overhead when taking into account personalized anonymity constraints compared to the performance measured by the original implementation of To et al.. Our implementation is tested using the classical adult dataset of UCI-ML [59] enhanced with a k_i privacy parameter for each tuple.

4.6.1 Experiments Platform

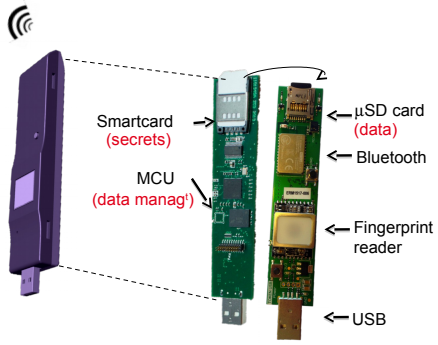


Figure 4.9: TDS characteristics [51].

The performance of the ^{k_i}SQL/AA protocol presented above has been measured on the tamper resistant open-source hardware platform shown in Figure 4.9. The TDS hardware platform consists of a 32-bit ARM Cortex-M4 microcontroller with a maximum frequency of 168MHz, 1MB of internal NOR flash memory and 196kb of RAM, itself connected to a μSD card containing all the personal data in an encrypted form and to a secure element (open smartcard) containing cryptographic secrets and algorithms. The TDS can communicate either through USB (our case in this study) or Bluetooth. Finally, a TDS embeds a relational DBMS engine, named PlugDB[7], running in the microcontroller. PlugDB is capable of executing SQL queries over the local personal data stored in the TDS. In our context, it is mainly used to implement the WHERE clause during the Collection phase of ^{k_i}SQL/AA.

[7]https://project.inria.fr/plugdb/en/

Our performance tests use the Adult dataset from the UCI machine learning repository [59]. This dataset is an extraction from the American census bureau database. We modified the dataset the same way of [43, 8]. We kept eight attributes to perform the `GroupBy` clause, namely `age`, `workclass`, `education`, `marital status`, `occupation`, `race`, `native country` and `gender`. Since our work is based on GROUP BY queries, we also kept the `fnlwgt` (*i.e.* final weight) attribute to perform an `AVG` on it. The final weight is a computed attribute giving similar value for people with similar demographic characteristics. We also removed each tuple with a missing value. At the end we kept 30162 tuples. Attributes `age` and `education` are treated as numeric values and others as categorical values. Since TDS have limited resources, categorical value are represented by a bit vector. For instance, the categorical attribute `workclass` is represented by a 8 bits value and its generalization process is performed by taking the upper node on the generalization tree given in Figure 4.10. The native country attribute is the largest and requires 49 bits to be represented.

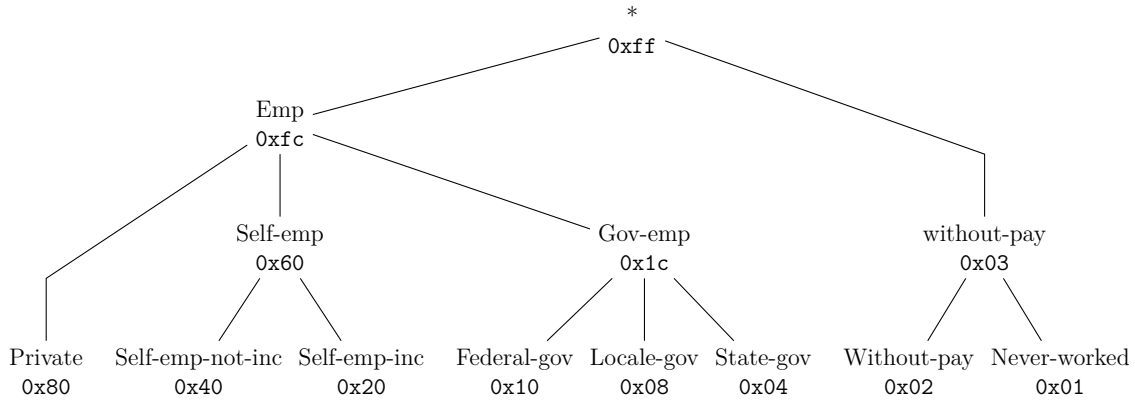


Figure 4.10: Generalization tree of `workclass` attribute.

4.6.2 Performance measurements

^{k_i}SQL/AA being an extension of SQL/AA protocol, this section focuses on the evaluation of the overhead incurred by the introduction of anonymisation guarantees in the query protocol. Then, it sums up to a direct comparison between ^{k_i}SQL/AA and original SQL/AA. To make the performance evaluation more complete, we first recall from [85] the comparison between

SQL/AA itself and other state of the art methods to securely compute aggregate SQL queries. This comparison is pictured in Figure 4.11. The Paillier curve shows the performance to compute aggregation in a secure centralized server using homomorphic encryption, presented in [33]. The DES curve uses also a centralized server and a DES encryption scheme (data are decrypted at computation time). Finally, SC curves correspond to the SQL/AA computation with various numbers of groups G (*i.e.* defined by `GroupBy` clause). This figure shows the strength of the massively parallel calculation of TDSs when G is small and its limits when G is really too big. We compare next the overhead introduced by our contribution to SQL/AA.

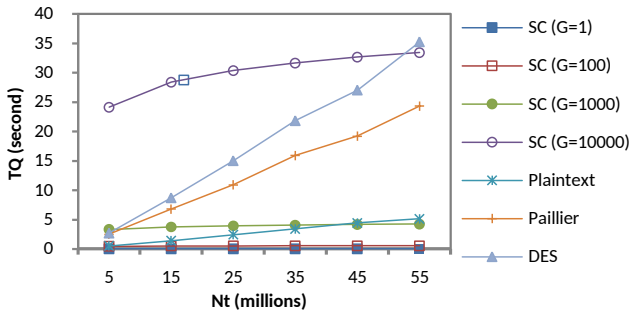


Figure 4.11: Performance measurements of SQL/AA and state of the art [85].

Categorical vs. numeric values. We ran a query with one hundred generalization levels, using first categorical, then numerical values. Execution time was exactly the same, demonstrating that the cost of generalizing categorical or numerical values is indifferent.

Collection and Filtering Phases. Figures 4.12 and 4.13 show the overhead introduced by our approach, respectively on the collection and filtering phases. The time corresponds to processing every possible generalization of the query presented in Figure 4.5, which generates the maximal overhead (i.e., worst case) for our approach. The *SQL/AA* bar corresponds to the execution cost of the SQL/AA protocol inside the TDS, the *data transfer* bar corresponds to the total time spent sending the query and retrieving the tuple (approximately 200 bytes at an experimentally measured data transfer rate of 7.9Mbits/sec), the *TDS platform* bar corresponds to the internal cost of communicating between the infrastructure and the TDS (data transfer excluded), and the *Privacy* bar corresponds to the overhead introduced by the ^{k_i}SQL/AA approach. All times are indicated in milliseconds. Values are averaged over the whole dataset (i.e. 30K tuples).

Collection Phase Analysis. The overhead of the collection phase resides in deciding how to generalize the tuple in order to comply with the local privacy requirements, and the global query privacy constraints. Figure 4.12 shows that our protocol introduces about a low overhead (between 0.29ms and 0.40ms). The variation is due to the number of generalization to be made. Most of the time is taken by the DBMS running in the TDS since it must access privacy preference on its NAND memory, which is slow. The time taken by the SQL/AA system come mostly from the time to retrieve data from the NAND memory which is *constant*. The same query could have been kept but we used the minimal query size to show this variation. Note that these performances are a bit different from the article [66] since we used more complicated queries on the TDS DBMS. Our contribution introduces an overhead under 10% of the overall time which we consider as a very reasonable cost.

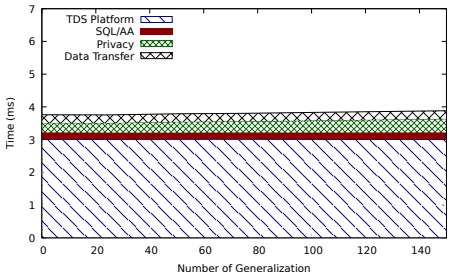


Figure 4.12: Collection Phase Execution Time Breakdown.

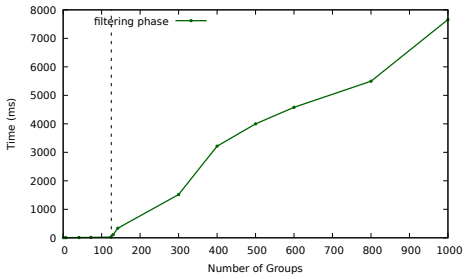


Figure 4.13: Filtering Phase Execution Time Breakdown.

Filtering Phases Analysis. Figure 4.13 shows the breakdown of the filtering phase execution time. The filtering phase takes place once the TDSs have computed all the aggregations and generalizations. The limited resources of the TDSs are bypassed by the SQL/AA system with the help of the (distributed) aggregation phase. Since every group is represented by one tuple, the TDS which computes the filtering phase receives a reduced amount of tuples (called G). To *et al.* have shown that the SQL/AA protocol converges if it is possible for a given TDS to compute G groups during the aggregation phase. As this is the number of tuples that will be processed during the filtering phase, we know that if G is under the threshold to allow its computation via the distributed aggregation phase, then it will be possible to compute the filtering phase with our improved protocol. To make performance measures,

the cortex-M4 has a 24bit system timer giving the possibility to measure durations lower than 100ms. Since computations now use NAND access, the time is too high to give a comparison between the SQL/AA system and the ^{k_i}SQL/AA system. The dashed line shows the limit where resorting to NAND is mandatory.

Aggregation phase Analysis. Figure 4.14 shows the performance measures of the sorting algorithm used in the aggregation phase. After the aggregation phase, the number of tuples the result contains is the number of different groups made by the `GroupBy` clause. We vary the number of groups to increase or decrease the overall time taken by each step of the merge algorithm. The limit of the SQL/AA system is shown by the dashed vertical line. The system was unable to compute a query with more groups than this limit. The merge algorithm permits to overcome this limit. The performance measure was done on one TDS. Since each step of the merge sort can be distributed, the maximum time of each partition merge was kept and summed between the different steps to get the time showed by the figure 4.14. The performance of the swap merge drastically decreases when the data to aggregate cannot fit in the RAM of the TDS and needs to be swapped on the NAND memory.

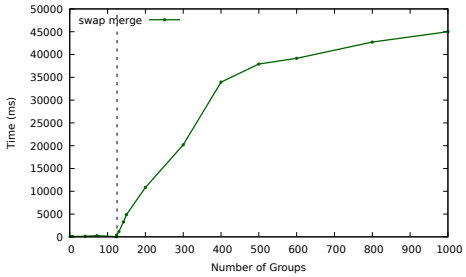


Figure 4.14: Aggregation Phase Execution Time Breakdown.

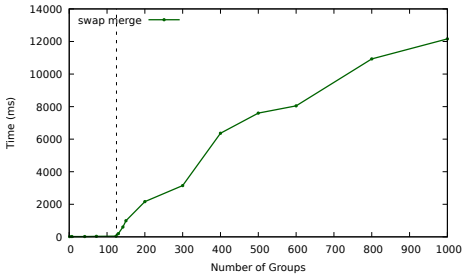


Figure 4.15: TDS Availability Required By The Aggregation Phase.

One of the weaknesses of the swap merge algorithm is the time a TDS must be available to merge two partitions. This time is directly related to the number of distinct groups generated by the `GroupBy` clause. It is important to measure this time. The maximum time a TDS must be available to compute the aggregation phase is illustrated by the figure 4.15. It was obtained by keeping the highest time between the sends of the message from the SSI and the receiving of the result by the SSI on the whole aggregation phase. The

time increases drastically when the TDS needs to use the NAND memory.

4.7 Conclusion

While privacy-by-design and user empowerment are being consecrated in the legislations regulating the use of personal data (e.g., EU General Data Protection Regulation), time has come to put these principles into practice. This chapter tackled this objective by proposing a novel approach to define personalized anonymity constraints on database queries. The benefit is twofold: let the individuals control how their personal data is exposed ever since collection time and, by this way, provides the querier with a greater set of consenting participants and more accurate results for their surveys. Moreover, we proposed a fully decentralized and secure execution protocol enforcing these privacy constraints, which avoids the risk of centralizing all personal data on a single site to perform anonymisation. Our experiments showed the practicality of the approach in terms of performance. Finally, this chapter showed that different semantics can be attached to the personalized anonymisation principle, opening exciting research perspectives for the data management community.

Chapter 5

Formalisation and Experimentation of Personalised $k-$Anonymity

In this chapter, we propose and experiment a new anonymisation model, called k_i-anonymity, which allows users to define their own value of k in the context of $k-$anonymity. We discuss several scenarios to cases where personnalization in anonymisation makes sense. We then present the model, and how to build an experimental dataset with k_i information. We discuss how current heuristics devised for $k-$anonymity must be adapted to solve the more complex problem of k_i-anonmity. We then show through a thorough experimentation on the classical adult dataset how k_i-anonymity impacts the quality of anonymisation results from the utility perspective.

5.1 Introduction

The anonymisation of personal information is an important task today. With the advent of *Smart-Disclosure* and *Open Data* the amount of collected data has exploded. The concept of *Big Data* is emerging giving new perspectives for statistics studies. Nowadays, data anonymisation is required before publishing. There are many ways to anonymised data as presented in section 3.2. While some statiscal institutes and industrial companies prefer the use of supervised methods such as the $\mu-$argus [42] toolkit or sdcMicro [84] R package, others prefer the use of partition-based methods such as $k-$anonymity and $\ell-$diversity or differential privacy. Concurrently, the GDPR [75] tries to push the possibility to empower users giving them more control over their data. However, most of the algorithms made to anonymise users' micro-data propose uniform anonymisation (*i.e.* based on a uniform security parameter) : no personalization exists. Only a small amount of these methods propose to empower users by giving them the possibility to manipulate the way their micro-data are anonymised. In the previous chapter, we studied how a querier can parameterize the way that users' personal data are generalized by controling generalization hierarchies and letting users specify the anonymisation level they need to participate, and using these information as input to guide generalization algorithms. In this chapter, we now consider the personalization of the privacy parameter itself, and how this will impact existing algorithms and heuristics.

Existing partition-based methods proposing personalisation can be separated into two approches. The first approach consists of methods offering users the manipulation of the granularity of the partition they will be in (*e.g. Clique-Cloak* algorithm). Unfortunately, all these methods are restricted to a certain type of data they take into account. They all generalize spatiotemporal data which are commonly used by location based services. However, in most cases, the data to be anonymised is much more varied and privacy-preserving methods must deal with many types of attributes (*e.g.* nominal, numerical, ordinal, categorical, ...). The second approach consists in modifying the way sensitive attributes are shared. Some solutions using *guarding nodes* allow the user's sensitive attributes to be generalized. The goal of methods such as $k-$anonymity is to keep sensitive information their integrity. Since those algorithms modify sensitive information, they do not have the same objectives as $k-$anonymity. Yet the personalisation of users'

privacy-preserving security and confidentiality is put forward by security institutes and legislators such as the GDPR [75]. Despite this interest, to the best of our knowledge, no one has studied the personalisation of partition-based privacy preserving methods regarding the granularity specified by their uniform parameter.

This chapter tackles the problem of the personalisation of $k-$anonymity by users (*i.e.* each user u_i can chose their own k_i value) to give them more control over their data and potentially improving the data quality. Instead of proposing yet another heuristic computing $k-$anonymity, this chapter proposes adaptations of existing $k-$anonymity heuristics. Three heuristics to be adapted have been chosen by their different scalability characteristics. One heuristic with a high scalability but low efficiency in term of data quality, a second heuristic with lower scalability but better efficiency and another one moderately scalable and moderately efficient. We also show that the adaptions of these heuristics do not complexify the algorithms. Another important objective of this chapter is to discuss the correlation between the privacy concern expressed by individuals and their personal data. As specified in section 3.1, privacy concern is correlated to some micro-data such as income, nationality or age. Since those correlations may affect the solution of personalised $k-$anonymity, experiments are done while simulating those correlations.

This chapter is organized as follows. We begin with giving several scenarios to illustrate use cases where personnalization in anonymisation makes sense 5.2. We then present the model, and how to build an experimental dataset with k_i information 5.3. We discuss how current heuristics devised for $k-$anonymity must be adapted to solve the more complex problem of k_i-anonymity in Section 5.4. We then show through a thorough experimentation how k_i-anonymity impacts the quality of results from the utility perspective 5.5.

5.2 Problem Statement and Scenarios

This section presents the problem statement and the different scenarios personalized privacy methods should take into account.

5.2.1 Problem Statement

This chapter tackles the problem of giving the possibility to users to specify their own privacy parameter used by $k-$anonymity. Following those views, the problem consists, also, to compute an anonymised dataset for data publishing ensuring the privacy parameter of each users.

5.2.2 Scenarios

To understand possible use cases in which personalised $k-$anonymity can be used, we propose different scenarios taking advantage of this new approach.

Scenario 1

The first scenario is pretty basic. It gives users, the possibility to determine "how much" they want to be anonymised. They can increase or decrease their privacy constraint (*i.e.* the k security parameter). Such scenarios are easy to understand in geolocation data, such as the anonymity circle defined by Chatzikokolakis et al. in the Location Guard add-on, based on geo-indistinguishability [5].

Scenario 2

Users have the possibility to gain an advantage by reducing their privacy constraint. They will get a greater advantage the more they decrease the k they want. To illustrate this scenario, we can use mobile application. For now, mobile applications use advertising and sell personal data as their business model. An application could propose to the user to lower the number of advertisments he sees by reducing his privacy constraint. By doing so, the data sold by the application would have a higher value which would lead to reducing advertisements seen by the user.

Scenario 3

The data controller, which is in charge of deciding how data is processed, wants to apply a $k-$anonymity algorithm on the data he is in charge of. A small amount of records want/need to have higher privacy and thus a higher k parameter. The data controller has several choices. The first is to compute $k-$anonymity with the highest k. This will clearly deteriorate

a lot of information of many records. A second choice would be to remove records with higher privacy needs and compute $k-$anonymity on a lower k. Finally, the third choice that we advocate would be to compute personalized $k-$anonymity to keep the information of both record groups (*i.e.* with and without higher privacy needs).

5.3 The k_i-anonymity model and linked concepts

5.3.1 k_i-anonymity definition

In this section, the personalized $k-$anonymity and used metrics are presented. Personalized $k-$anonymity aims in empowering users on the specific security parameter. Indeed, most of the related works about personalized $k-$anonymity give the possibility to user to specify how their sensitive attributes are shared. Instead, we propose the personalization of the security parameter k. We will denote this concept as k_i-anonymity.

Definition 8 (k_i-anonymity)**.** Let $\mathcal{D}$ be a table of records, k_i denote the k chosen by the user $i \in \mathcal{D}$. Let QID_i denotes the quasi-identifier of this user. The table $\mathcal{D}$ is k_i-anonymous iff $\forall i \in \mathcal{D}$, $|\{QID_j \mid j \in \mathcal{D}, QID_j = QID_i\}| \geq k_i$ (with $|S|$ denotes the cardinality of the set S).

Rationale

Table 5.1 shows an example of a k_i-anonymous table (based on the table 3.3). The idea is not to group users with the same privacy constraint together but to group individuals to achieve best utility, while respecting the k_i constraint. The general approach of constrained optimization is thus similar to k-anonymity, but constraints vary from class to class depending on the members of a class, which means that it is much more difficult to propose heuristics, than in the case where there is only one parameter for all classes. For instance, one big difference with $k-$anonymity is that creating the maximum number of equivalence classes possible does not mean having the best data quality: we might be grouping together individuals that are very different, simply to optimize the k_i constraints, by putting the least possible individual in each group, regardless of their similarity. Such a problem does not appear with $k - anonymity$.

Quasi-identifier		Sensitive	k_i
ZIP	Age	Condition	
1402*	$[25, 35]$	Cancer	2
1402*	$[25, 35]$	Heart Disease	2
1402*	$[25, 35]$	Cancer	2
141**	$[38, 45]$	Heart Disease	3
141**	$[38, 45]$	Viral Infection	3
141**	$[38, 45]$	Viral Infection	2
141**	$[38, 45]$	Cancer	3
1402*	$[50, 70]$	Viral Infection	2
1402*	$[50, 70]$	Cancer	2

Table 5.1: Example of fictitious medical dataset k_i-anonymous.

5.3.2 Metrics for k_i-anonymity heuristics

Many current $k-$anonymity algorithms are based on heuristics that use metrics to take their decisions. The example just discussed in the previous paragraph illustrates why metrics such as the discernability metric or the normalized average equivalence class size metric do not fit the problem of k_i-anonymity. Using them would consist to first partition records by their constraints regardless to their similarity. As shown by table 5.1, grouping records with the same constraint regardless of their similarity to create an extra group can lead to more information loss! For example here, a new equivalence class can be created with the third and the sixth records which would increase global information loss.

In order to represent information loss, we use a metric which is better adapted to the fact that each group will have a specific k_i parameter: the diameter based metric (already presented in section 3.4):

$$DBIL(e) = |e| \cdot diameter(e) \tag{5.1}$$

where e is an equivalence class of records in $\mathcal{D}^*$. The diameter of an equivalence class is computed by the highest distance between two records in the equivalence class. The distance between two records is the Manhattan distance with attributes as coordinates. The distance of numeric attributes consists of the substraction of the greatest value by the lowest and the distance of two pieces of categorical data is the height of the smallest subtree

containing each value in the taxonomy tree $\mathcal{T}$ of the attribute. This information loss metric can be seen as the highest Manhattan distance of the equivalence class multiplied by the cardinality of the equivalence class. The distance of two records on an attribute is normalized to ensure each attribute has the same importance but a factor can be added to give more importance to some attributes (*i.e.* some attributes may be more important than others, see 4.3.3). To compute the overall information loss of the anonymised dataset $\mathcal{D}^*$, the sum of equivalence classes information loss can be computed.

Other metrics could also be used to guide a k_i-anonymity heuristic, so long as they take into account the similarity of groups.

5.3.3 Personnalized privacy datasets

In order to test the concept, we need data. Unfortunately, to our knowledge, no dataset exists where individuals have been asked to chose a certain degree of protection when anonymising their data. For instance, we contacted K. Chatzikokolakis to see if they had information about the privacy parameter chosen by users using their Location Guard plug-in, but indeed, this parameter was not collected (which is normal since this is in some sense sensitive information). Another experiment was run by Katsouraki [47] in order to measure how individuals behave when sharing personal information depending on their perception of information security, but was not formalized in terms of k_i values. Thus, we discuss here an existing study by Acquisti et al. and how we can extrapolate it in order to build a *realistic dataset* to be used in personalized anonymisation.

Individual Privacy concern

We discuss the impact of the work of A. Acquisti and J. Grossklag already discussed in the section 3.1 by presenting a discussion about the distribution of privacy constraints (*i.e.* privacy parameter k) based on their study. The study of A. Acquisti and J. Grossklags shows how the privacy concern of individuals varies regarding the category of targeted personal information. Table 5.2 is a part of the survey result of A. Acquisti and J. Grossklags paper [2].

They identified four groups of people differing by their privacy concerns. The conservative group is composed of people having a high privacy concern.

	General privacy concern	Data about personal profile	Data about professional profile	Data about sexual and political identity
High concern	53.7 %	0.9 %	11.9 %	12.1 %
Mediumconcern	35.5 %	16.8 %	50.8 %	25.8 %
Low concern	10.7 %	82.3 %	37.3 %	62.1 %

Table 5.2: Privacy concern in function of data category [2].

In the moderate group, they merged two groups with a medium privacy concern. The liberal group is composed of people with low privacy concern.

We consider in this thesis the distribution of privacy concern already studied by Acquisti et al. to represent the privacy constraints individuals will choose.

Data correlation

A. Acquisti and J. Grossklags have shown that the privacy concern was correlated to the income. This is important, since we do not simply want to apply the privacy concern distribution blindly: since we are going to be using the well know adult dataset [59], if we are able to find a correlation between privacy concern and some of the attributes present in this dataset, then our experiment will be more realistic.

Indeed, the existence of a correlation is consistent with others studies [48, 15, 13] which show that young adults are less concerned by privacy than older. It means that in a dataset, clusters of similar records could show different distributions over conservative, moderate and liberal groups. Figure 5.1 illustrates the correlation between the age and income on the privacy concern over a fictious dataset. In this figure we can see two different clusters. The *concerned* cluster would have a distribution over conservative, moderate and liberal groups near the general privacy concern of table 5.2 (*i.e.* 53.7% of conservatives, 35.5% of moderates and 10.7% of liberals) when the *concernless* cluster would have a distribution more like the data about personal profile (*i.e.* 0.9% of conservatives, 16.8% of moderates and 82.3% of liberals).

This correlation shows that even with a high majority of individuals with a high privacy concern, there will be some clusters of records with lower overall privacy concern in which information loss will clearly be reduced through the

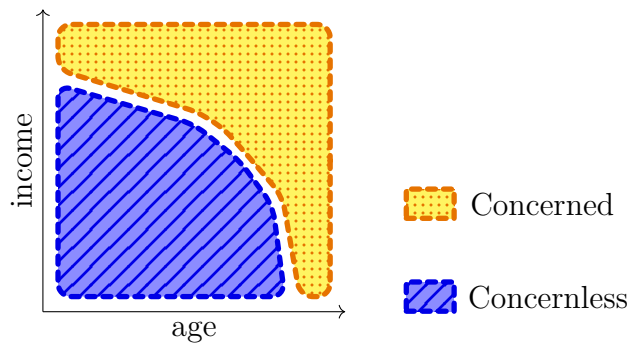


Figure 5.1: Correlation between data and privacy concern over a fictitious dataset.

use of a k_i-anonymity approach, which will lead to gains in scenario 3 for instance.

5.4 Personalised $K-$Anonymity Heuristics

In this section, three heuristics that have been adapted to the problem of personalized $k-$anonymity are presented. To implement heuristics for the k_i-anonymity, we choose to adapt three heuristics presented in section 3.4 which this section will highlight in more details. The first one is the *Mondrian* (*i.e.* median partitionning) algorithm proposed by K. Lefevre et al. [54, 55] which gives a low data quality but has a high scalability because of its low complexity $O(n \cdot log(n))$. The second heuristic gives a better quality but is much less scalable since its complexity is in $O(n^2 \cdot k)$. This heuristic is called the *Greedy k-member clustering* proposed by J. Byun et al. [14]. The last one is a trade off between quality and scalability comparing to the two others with a complexity in $O(n^2)$. It is called the *MDAV* algorithm and was proposed by J. Domingo-Ferrer and V. Torra [21].

5.4.1 Mondrian adaptation

The idea of the Mondrian heuristic is to cut the dataset by its median on the wider (*i.e.* less generalized) attribute. This is equivalent to a top-down algorithm used with local recoding and it takes advantage of the *divide-and-conquer* approach. The median can be found after the computation of a

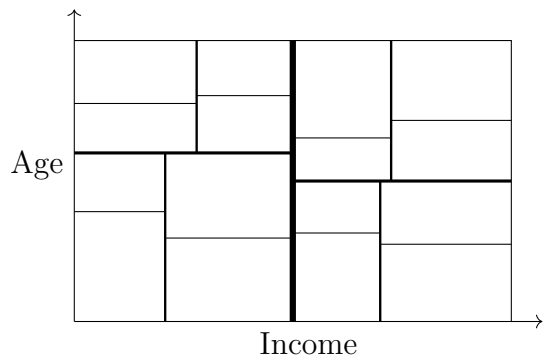


Figure 5.2: Example of partitionning by Mondrian heuristic

frequency set which highlights the values' occurences. This frequency set also permits to detect when a categorical attribute cannot be specialized. The two partitions obtained are then considered as new dataset and the method is reiterated on those partitions. The algorithm stops when the cardinality of partitions is lesser than $2 \cdot k$ (*i.e.* the partition cannot be cut again). The idea of the Mondrian adaptation is to set the k value of a partition to the maximum k_i constraint of each record in the partition. By doing so, the heuristic can produce equivalence classes of smaller size than using the highest constraint as a security parameter. Figure 5.2 illustrates how a dataset can be partitioned by Mondrian. It is important to note that the algorithm has a complexity of $O(n \cdot \log n)$. Since the Mondrian heuristic cuts a partition in two, the highest privacy constraint of the two partitions can be computed during the partitionning process.

5.4.2 MDAV and Greedy $k-$member clustering adaptations

The MDAV heuristics and the Greedy $k-$member clustering (called Greedy $k-$member in the rest of the manuscript for clarity) are a bit similar. The MDAV heuristic computes the centroid c of the dataset $\mathcal{D}$ and does not necessarily belong to $\mathcal{D}$. It then takes the furthest record r_1 from c and the furthest record r_2 from r_1. Two equivalence classes e_1 and e_2 are built by taking the $k-1$ nearest records from r_1 (*resp.* r_2) creating equivalence class e_1 (*resp.* e_2). Records from the two equivalence classes are removed from the dataset and the operation is reiterated until no records are left. If less than

k records are remaining, the records are merged to the nearest equivalence classes. On the other hand, the Greedy $k-$member does not compute the centroid of the dataset. It takes a random records r_i and builds an equivalence class from the furthest record r_1 from r_i. Although they do not begin the same way, MDAV and Greedy $k-$member try to create equivalence classes beginning with the edge of the dataset. The biggest difference of MDAV and Gredy $k-$member is the way they build the equivalence class after selecting a record from the edge of the dataset. Instead of taking the $k-1$ nearest records from r_1, the Greedy $k-$member fill the equivalence class e which r_1 belongs with the records r that minimizes the following formula:

$$IL(e \cup r) = |e \cup r| \cdot diam(e \cup r) \tag{5.2}$$

with $e \cup r$ the equivalence class e merged with the record r. Like the MDAV heuristic, when an equivalence class is created, records from it are removed from the dataset and the process is reiterated. The adaptation of those two heuristics to the k_i-anonymity is pretty straightforward. While building equivalence classes, the number of records to put in those depends of the highest privacy constraint associated to records from those equivalence classes. This highest constraint can be kept up-to-date while building the equivalence class e (*i.e.* $e.k \leftarrow max(e.k, k_r)$ when appending r to e). This way, the complexity of the two heuristics does not increase when adapted to k_i-anonymity.

5.4.3 $K-$member greedy improvement

The $K-$member greedy selects the record r which adds the least information loss to the equivalence class e before publishing. It consists in minimizing the following $diff$ value by selecting the closest record r:

$$diff = (|e| + 1) \cdot (diam(e \cup r)) - |e| \cdot (diam(e)) \tag{5.3}$$

The disavantage of this value is that it does not avoid the absorption of low constrained records by highly constrained records. Since the heuristic does not remove any records, the computation of the estimation of equivalence class e information loss on the $diff$ equation 5.3 is wrong. Since e will have a cardinality of at least k, the equation should be:

$$diff = max\,(|e| + 1, k) \cdot (diam(e \cup r)) - max(|e|, k) \cdot (diam(e)) \tag{5.4}$$

It is not inconvenient when computing $k-$anonymity since all equivalence classes have the same constraint but when used on k_i-anonymity, the computation of the $diff$ value should take into account the different constraints of records:

$$diff = max\,(|e| + 1, k_e, k_r) \cdot (diam(e \cup r)) - max(|e|, k_e) \cdot (diam(e)) \quad (5.5)$$

with k_e the greatest constraint in e and k_r the constraint of r. Using this new equation permits to have equivalence classes fitting better to the privacy constraints of their records.

5.5 Experimenting k_i-anonymity

This section conducts a thorough experimentation of personnalized k_i-anonymity, throught the study of heuristics adapted to our new model, and a comparison to their original version. Measures show the evolution of information loss when varying some parameter. Our performance tests use the Adult dataset [59] the same way described in section 4.6 (*i.e.* using six categorical and two numerical attributes and removing records with empty values). To determine the privacy constraints of users, we use the model of A. Acquisti and J. Grossklags [2] with some examples reminded by table 5.2. We considered three groups with different levels of privacy constraints, conservatives having the highest constraint k_c, moderates have a medium constraint k_m and liberals have the lowest constraints k_l. Values for k_l, k_m and k_c were chosen on the same scale of the INSEE (*i.e.* the french *National Institute of Statistics and Economic Studies*) methodology. They chose $10-$anonymity and $3-$diversity to protect medical data [11], a $5-$anonymity for the *Labour Force Survey* and a $3-$anonymity for a survey about *theft, violence and security* [10]. Experiments where done with two different types of privacy constraints, one with *low* privacy constraints ($k_l = 3$, $k_m = 5$ and $k_c = 7$) and the other with *high* privacy constraints ($k_l = 5$, $k_m = 7$ and $k_c = 10$). Note that tuples selected as conservators (*resp.* liberals and morderate) where not the same when changing the privacy constraints from *low* to *high* (*i.e.* a shuffle of privacy constraints and adult dataset is done at each experiment).

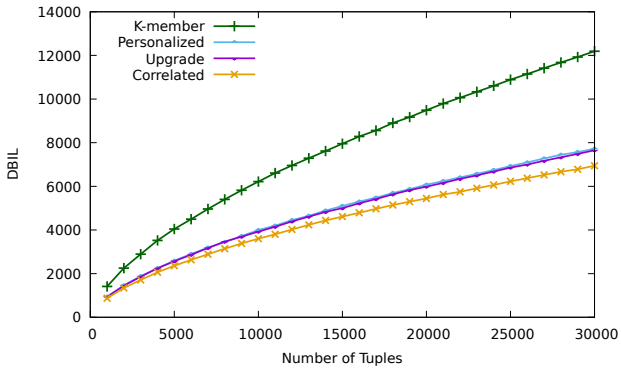
The first part shows the differences between original heuristics and their adaptions to k_i-anonymity. The second part exhibits the influence of the variation on the distribution of the three groups (*i.e.* liberals, moderates

and conservatives). Then, we present the influence of how much the distance on privacy constraints between conservatives, moderates and liberals impacts information loss. After that, we show the impact of the suppression of conservatives to achieve k_m anonymity and compare it with the use of k_i-anonymity instead. Finally, we experiment on the correlation of privacy constraints and quasi-identifiers values and discuss it.

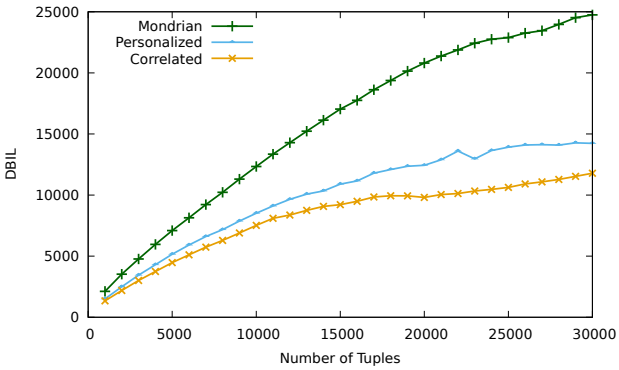
A taxonomy trees as been produced for each categorical value. The workclass taxonomy tree is given by figure 3.10 above to explain metrics about taxonomy trees. Only taxonomy trees about `gender` and `race` were trees of height 1 (*i.e.* the generalization of these value comes with the suppression of these).

5.5.1 Heuristics comparison with their original

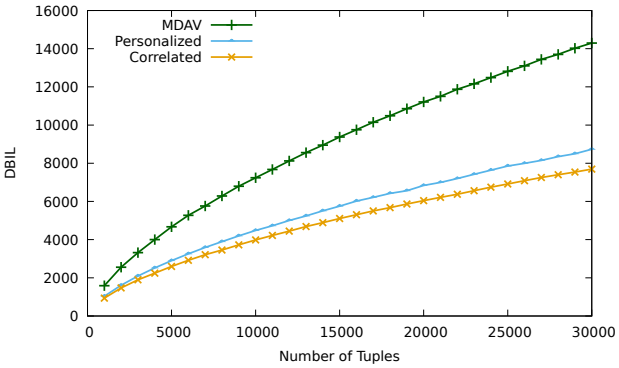
Figure 5.3 shows the information loss (using $DBIL$ metric) of $k-$anonymity computed by the three different heuristics and compares it with their adaptation to k_i-anonymity. The variable used in the $x-$axis is the number of records taken. Those records were selected randomly from the aldult dataset after the modifications specified in section 4.6. In this experiment, conservatives were given a security parameter of $k_c = 7$, moderates were given a $k_m = 5$ and liberals a $k_l = 3$. The distributions of the conservatives, moderates and liberals were given by the personal profile (see table 5.2) of the paper of A. Acquisti and J. Grossklags [2] and privacy constraints were given randomly following this distribution. The curves labeled «*Personalized*» represent the information loss generated by the adaptation of the heuristics (labeled by their name) to k_i-anonymity. Original heuristics were used by selecting the highest privacy constraint $k_c = 7$ as security parameter. We discuss the curves labeled «*Correlated*» in section 5.5.5 dedicated to the correlation of personal information and privacy concern. The first obversation we can do is the efficiency of those three heuristics. The Mondrian algorithm is the one generating the most information loss. This is not surprising since this algorithm is a median partitionning in $O(n \cdot \log n)$. On the other side, the Greedy $k-$Member and MDAV algorithms are pretty close. The MDAV heuristic generates a bit more information loss than the Greedy $k-$Member which also corresponds to their difference in term of complexity. The information loss grows slower when increasing the number of records used because increasing the number of records in a dataset without modifying the domain of attributes is equivalent to increasing the density of



(a) Greedy $k-$Member

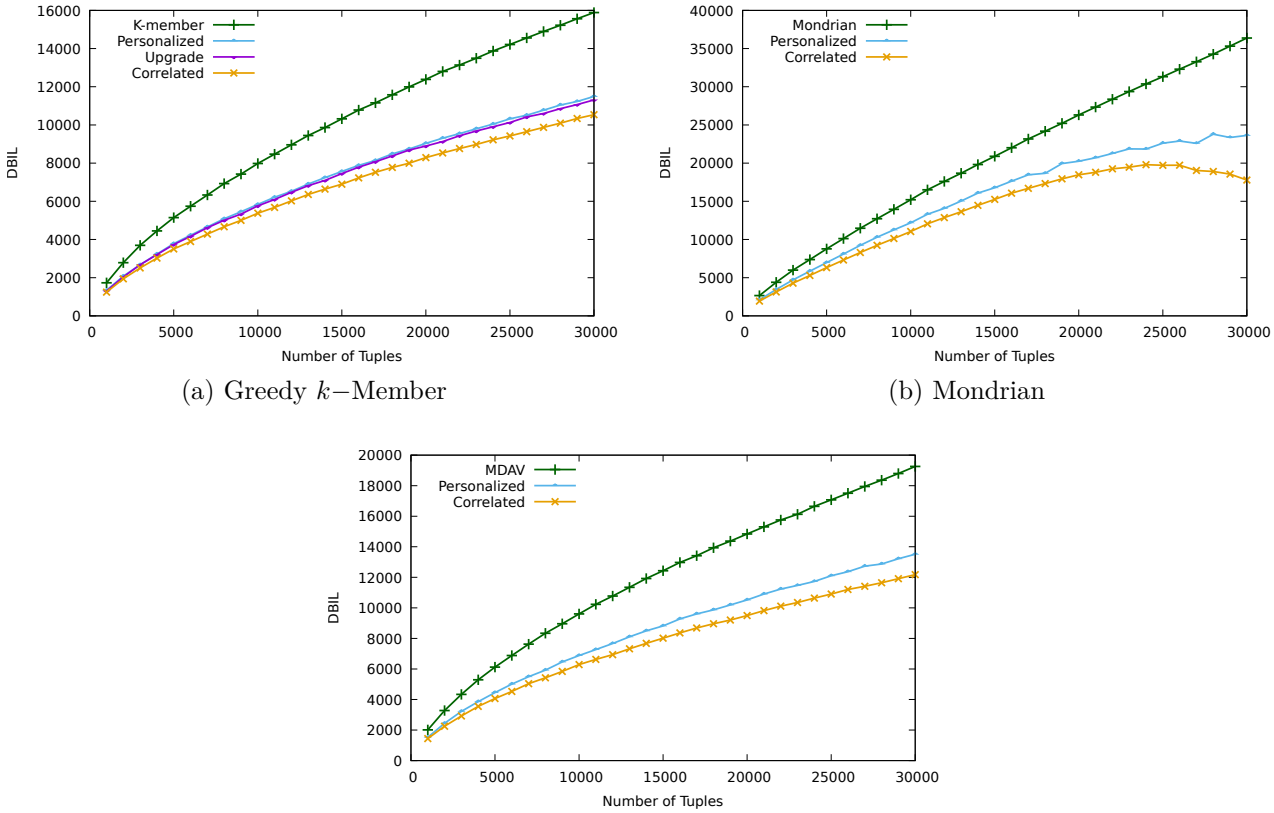


(b) Mondrian



(c) MDAV

Figure 5.3: Information loss of heuristics with the personal profile distribution and $k_l = 3$, $k_m = 5$ and $k_c = 7$



(a) Greedy $k-$Member

(b) Mondrian

(c) MDAV

Figure 5.4: Information loss of heuristics with the personal profile distribution and $k_l = 5$, $k_m = 7$ and $k_c = 10$

the dataset and therefore increasing the similarity between records and their nearest neighbors. As we can see, personalized $k-$anonymity generates less information loss than each original heuristic. Since k_c-anonymity respects all privacy constraints, by construction, k_i-anonymity should not produce more information loss for scenario 2 (*i.e.* users are able to reduce their constraint to access some services). The purple curve labeled *Upgrade* represents the modification of the way information loss estimation is computed by the Greedy $k-$Member algorithm presented in section 5.4.3. We can see that this modifaction decreases a bit the information loss generated.

Figure 5.4 shares the same characteristics as figure 5.3 but with different

security parameters. In this figure, conservatives were associated with a security parameter $k_c = 10$, moderates $k_m = 7$ and liberals $k_l = 5$. By comparing the two figures, we can see that the gain of quality is lower when constraints increase. It is due to the fact that the probability records with a high constraints *absorb* records with lower constraint increases (*i.e.* low constrained records are put in high constrained equivalence classes). This probability increases because having larger equivalence classes tends to put all kinds of records (including those with high constraints) in those equivalence classes. This causes a loss of the potential data quality gain of k_i-anonymity because this gain happens when equivalence classes with no conservatives are created. It is important to note that the use of original heuristics induces an increasing information loss because they were parameterized with the conservors' security parameter k_c. By design, adaptations of those heuristics to k_i-anonymity cannot induce a greater information loss when comparing to originals parameterized by k_c.

The frequencies over conservatives, moderates and liberals also have an impact on the the information loss when using k_i-anonymity as we will see in next section.

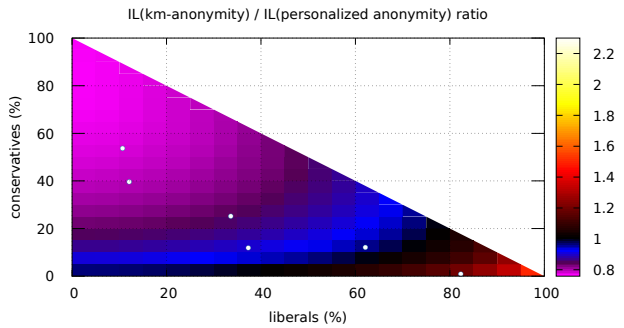
5.5.2 Impact of frequencies high/low constraint groups

Larger versions of figures presented in this section can be found in the appendix. The appendix aims to make these figures more readable.

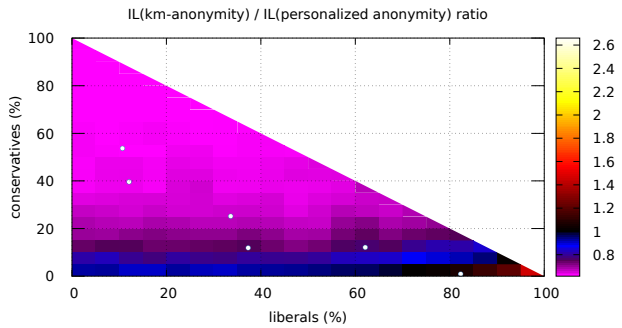
Figure 5.5 is a 3D plot of k_i-anonymity by varying the frequency of conservatives f_c on the $y-$axis and the frequency of liberal f_l on the $x-$axis. Since the frequency of moderates is computed as follows:

$$f_m = 100 - (f_l + f_c), \tag{5.6}$$

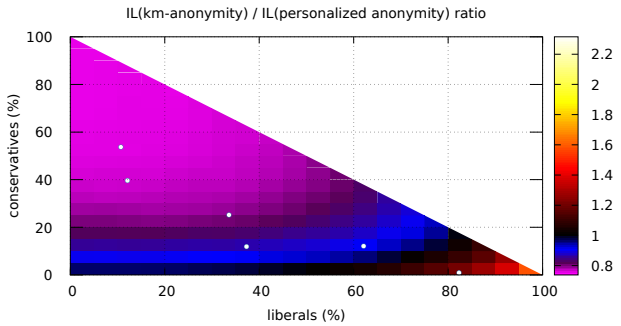
f_m is higher when near to the origin and lower otherwise (*i.e.* 100% at the origin and 0% on the hypothenuse of the graph). The third dimension is represented by color and shows the ratio of information loss generated by $k-$anonymity heuristics with $k = k_m$ over information loss generated by their k_i-anonymity version (*i.e.* $DBIL(k-\text{anonymity})/DBIL(k_i-\text{anonymity})$). The security parameter given to the conservative group is $k_c = 7$, the moderate group receives $k_m = 5$ and the liberal one $k_l = 3$. Data for those graphs was computed with the adaptation of the Greedy $k-$Member heuristics with the improvement of its estimation of information loss shown by figure (a),



(a) Greedy $k-$Member

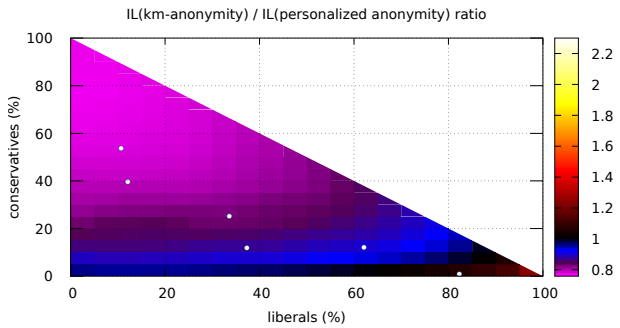


(b) Mondrian

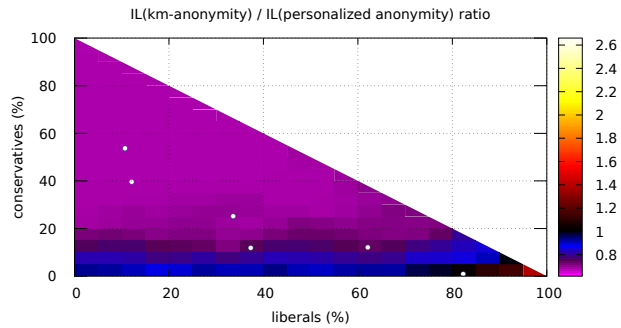


(c) MDAV

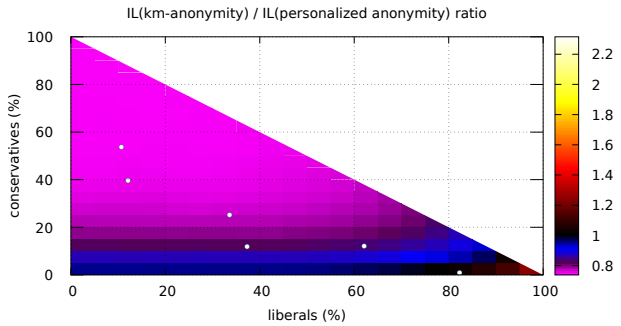
Figure 5.5: Comparison of k_i-anonymity and k_m-anonymity by varying f_l and f_c with $k_l = 3$, $k_m = 5$ and $k_c = 7$ (scenario 1)



(a) Greedy $k-$Member



(b) Mondrian

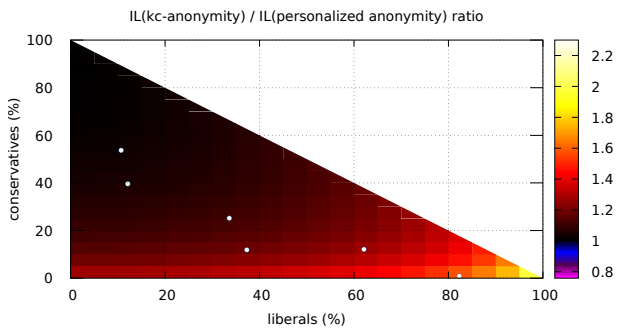


(c) MDAV

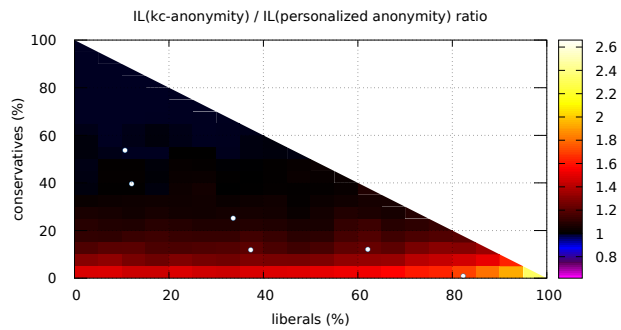
Figure 5.6: Comparison of k_i-anonymity and k_m-anonymity by varying f_l and f_c with $k_l = 5$, $k_m = 7$ and $k_c = 10$ (scenario 1)

the Mondrian heuristics shown by figure (b) and the MDAV heuristic shown by figure (c). The white dots on the graphs represent the different distributions given by the study of A. Acquisti and J. Grossklags reminded by table 3.1. Figure 5.6 shares the same characteristics as figure 5.5 but for $k_c = 10$, $k_m = 7$ and $k_l = 5$. Following the scenario 1, those figures show from which frequencies of conservatives, moderates and liberals, information loss would be higher with k_i-anonymity than the k_m-anonymity. Due to the fact that records with high privacy constraints absorb others records in their equivalence classes, the information loss quickly decreases when their frequency increases. In cold colors (*i.e.* from black to blue then purple), the information loss is lower when computing $k-$anonymity with k_m as security parameter (and so thus *not* respecting users constraints) than using k_i-anonymity. In hot colors (*i.e.* from black to red then yellow), the information loss is lower with k_i-anonymity than the $k-$anonymity. In scenario 1, the idea is to give the possibility to users to lower their privacy constraints but also to increase them. Based on the privacy concern table 5.2, the scenario 1 would reduce information loss for personal profile data but increase information loss for others data. For data about sexual and political identity, the information loss would be almost the same as the information loss generated by k_m-anonymity. This means that even if you give the possibility to users to specify their privacy constraints, the information loss would not increase that much for those kinds of data (*i.e.* sexual and political identity), which is a positive result of this study. For other cases, even if information loss would be worse when giving users the possibility to increase or decrease their privacy constraints, we assume it would be an acceptable tradeoff to offer more control to users over their data.

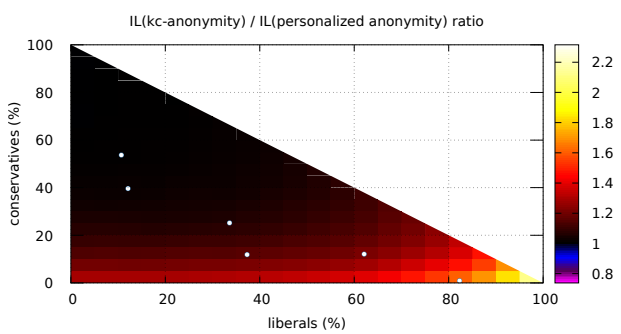
Figure 5.7 and the figure 5.8 share the same characteristics as the previous figures 5.5 and 5.6 but when comparing k_i-anonymity to k_c-anonymity (*i.e.* $k-$anonymity with $k = k_c$). In figure 5.7, $k_c = 7$, $k_m = 5$ and $k_l = 3$ while in figure 5.8, $k_c = 10$, $k_m = 7$ and $k_l = 5$. The figures show the impact of the reduction of users' privacy constraints on the information loss generated by the anonymisation process. Regarding the privacy concern table 5.2, almost every distribution would cause a decrease of information loss when k_i-anonymity is used. With the personal profile data, the information loss with k_c-anonymity would be about 1.6 times the information loss with k_i-anonymity when using the Greedy $k-$Member heuristics, about 1.8 times when using the Mondrian heuristic and about 1.6 times when using the MDAV heuristic with low contraints (*i.e.* $k_c = 7$, $k_m = 5$, $k_l = 3$). We can



(a) Greedy $k-$Member

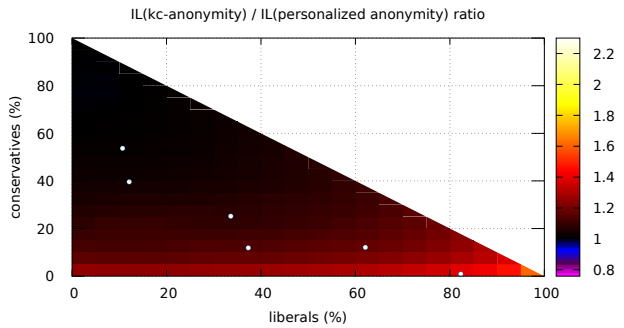


(b) Mondrian

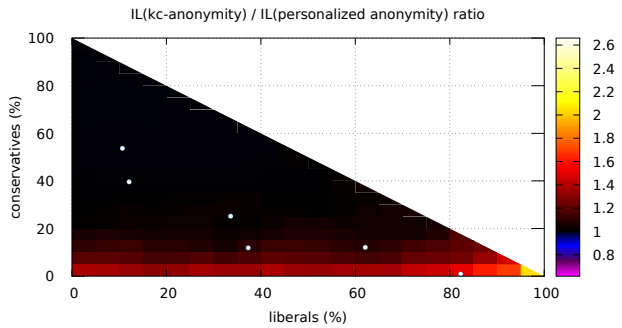


(c) MDAV

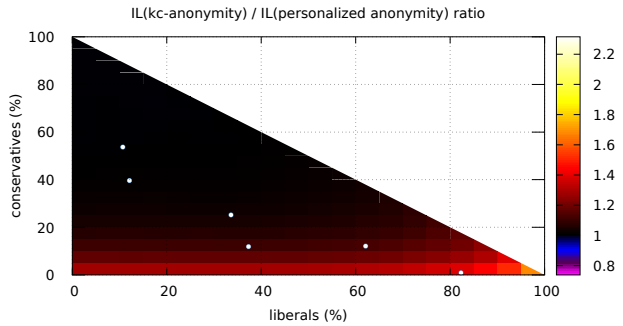
Figure 5.7: Comparison of k_i-anonymity and k_c-anonymity by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 3$, $k_m = 5$ and $k_c = 7$ (scenario 2)



(a) Greedy $k-$Member



(b) Mondrian



(c) MDAV

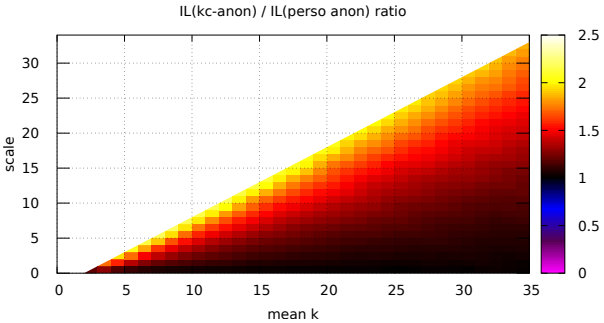
Figure 5.8: Comparison of k_i-anonymity and k_c-anonymity by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 5$, $k_m = 7$ and $k_c = 10$ (scenario 2)

also see that increasing the overall security parameter causes the reduction of information loss to be lower. Thus, increasing security parameters too much would give no advantage over computing k_c-anonymity. Regarding the scenario 2, a mobile application using advertising and statistics as business model could reduce the quantity of advertising function of the type of data and the frequency of users lowering their constraints.

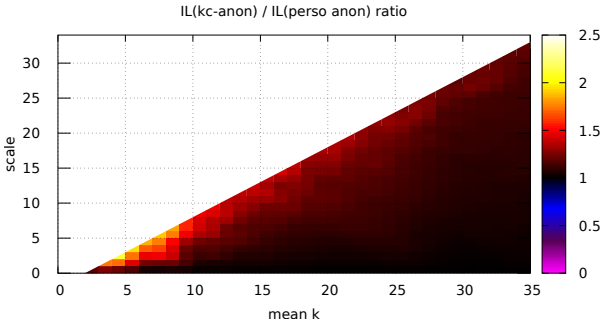
5.5.3 Impact of the scale between high and low constraints

Larger versions of figures presented in this section can be found in the appendix. The appendix aims to make these figures more readable.

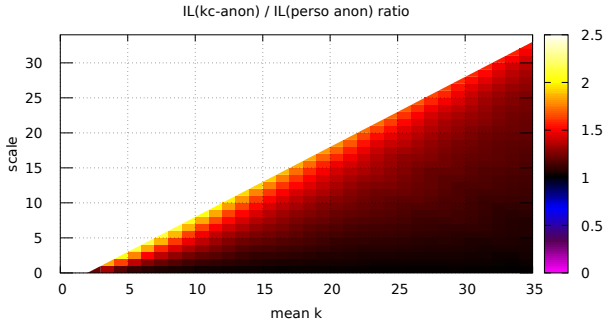
Figure 5.9 shows the ratio of information loss generated by k_c-anonymity over information loss generated by k_i-anonymity while varying the *scale* and k_m. The *scale* represents the space between privacy constraints (*i.e.* $k_l = k_m - scale$ and $k_c = k_m + scale$) and is represented by the $y-$axis. On the other hand, k_m is represented by the $x-$axis. Colors indicate how much the information loss decreases when using k_i-anonymity instead of k_c-anonymity. A higher value (*i.e.* yellow) is better (black means almost no improvement). The distributions of conservatives, moderates and liberals are set following the personal profile distribution of the privacy concern table 5.2. The figures show that the higher the scale, the more the gain of data quality provided by k_i-anonymity is important. This is due to the fact that conservatives are a small portion of records. When the scale is high the difference between the information loss of equivalence classes of size k_c and equivalence classes of size k_m or k_l is greater. Since the increase of overall security parameters (consists in going to the right on the graph) reduces the data quality gain, the figure shows also that the more the constraints are high (*i.e.* k_m) the lower the difference between k_c-anonymity and k_i-anonymity. This means that there is a limit were changing the scale would not affect information loss because of too high constraints. For instance, defining $k_l = 2$, $k_m = 15$ and $k_c = 28$ (*i.e.* scale of 13) gives a greater improvement that defining $k_l = 2$, $k_m = 35$ and $k_c = 67$ (*i.e.* scale of 33) even if the scale was maximised in both cases.



(a) Greedy $k-$Member

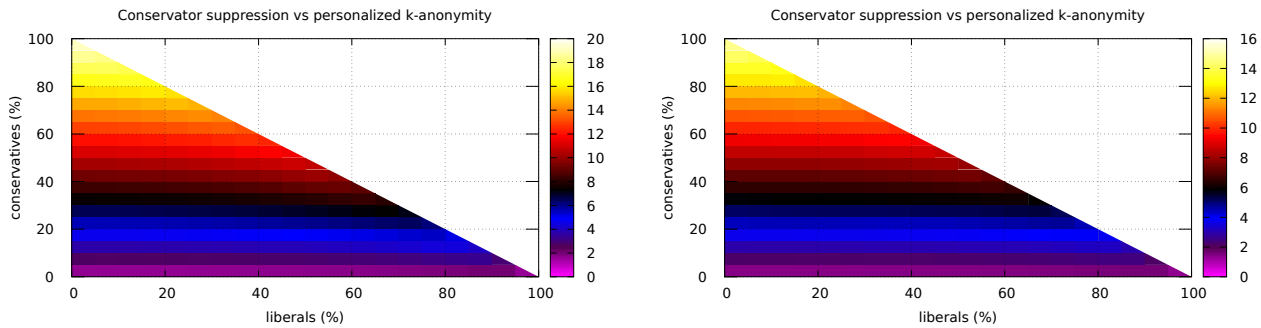


(b) Mondrian



(c) MDAV

Figure 5.9: Comparison of k_i-anonymity and k_c-anonymity by varying the scale and k_m on the personal profile distribution (scenario 2)



(a) $k_l = 3$, $k_m = 5$ and $k_c = 7$

(b) $k_l = 5$, $k_m = 7$ and $k_c = 10$

Figure 5.10: k_i-anonymity vs k_m-anonymity after removing conservatives, using Greedy $k-$Member (scenario 3)

5.5.4 Impact of the conservatives deletion

Larger versions of figures presented in this section can be found in the appendix. The appendix aims to make these figures more readable.

Figure 5.10 illustrates the scenario 3. The two figures differ from the overall security parameter (*i.e.* (a) with low contraints, (b) with high constraints). Results were computed using the Greedy $k-$Member heuristic. It shows the ratio between the information loss generated by k_m-anonymity after removing conservatives over the information loss generated by k_i-anonymity while varying frequencies of conservatives and liberals. The figure shows that the information loss is excessively high when removing conservatives. It is due to the fact that the penalty for the deletion of a record is much higher than the general information loss an equivalence class generates. For instance, the deletion of a record generates an information loss of 8 (*i.e.* number of attributes) while the information loss generated by records in an instance of Mondrian k_i-anonymity vary from 0 to 5.0274 (mean is equal to 1.6758). The information loss generated by equivalence classes is negligible compared to the information loss generated by the deletion of conservatives. Following the scenario 3, it would always be better to use k_i-anonymity than not taking into consideration conservatives data to allow the computation of k_m-anonymity. This shows that our approach is clearly positive in all cases for this kind of scenario.

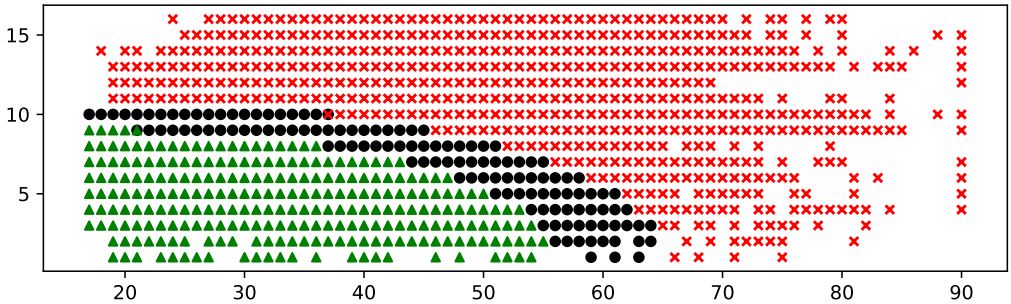
5.5.5 Impact of the correlation between data and privacy concern

Larger versions of figures presented in this section can be found in the appendix. The appendix aims to make these figures more readable.

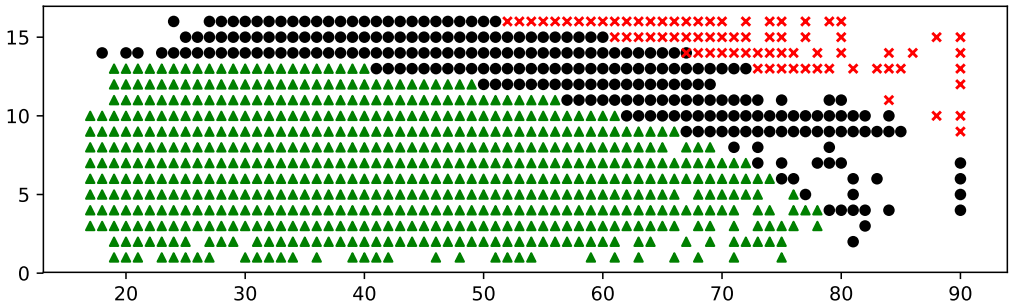
This section tackles the problem of k_i-anonymity with privacy concern correlated to some personal information. To experiment the correlation we selected attributes *age* and *education* as attributes correlated to privacy concern. Following distributions given by A. Acquisti and J. Grossklags, privacy parameters where given by projecting the dataset in the 2D plan with `age` as the $x-$axis and `education` as the $y-$axis then, the nearest records to the origin where given a low constraint. Figure 5.11 shows the attribution of privacy constraints in function of the distribution given by A. Acquisti and J. Grossklags. Figure (a) shows the distribution when taking the *general privacy concern* in consideration and figure (b) illustrates the attribution of privacy constraints while using *data about personal profile* distribution.

Figure 5.12 illustrates the ratio between the information loss generated when no correlations were taken into account over the information loss generated when the simulation of the correlation between privacy concern and `age` and `education` was used (*i.e.* $DBIL$(no correlation)/$DBIL$(correlation)). This ratio is represented by the colors, black means no improvement while yellow means the highest improvement. In this figure, the conservatives were given a constraint $k_c = 7$, the moderates $k_m = 5$ and the liberals $k_l = 3$. The $x-$axis illustrates the frequency of liberals and $y-$axis illustrates the frequency of conservatives. Since the reduction of information loss was not of the same magnitude for the three heuristics, the color ranges used were not the same for the different figures (a), (b) and (c). The first observation we can make is that the Mondrian heuristic is much more impacted by the correlation. It is not surprising since it tends to create larger equivalence classes which cause equivalence classes to have better chances to get a conservative in them when no correlation exists. However, when privacy constraints are correlated to some attributes, age and education in our case, then the partitionning of those attributes will gather conservatives (*resp.* moderates and liberals) together. It has the same effect on Greedy $k-$Member and MDAV heuristics.

Figure 5.13 shares the same characteristics as the previous one but the constraints are greater. The conservatives were given a constraint $k_c = 10$, the moderate $k_m = 7$ and the liberals $k_l = 5$. In the six figures, we can

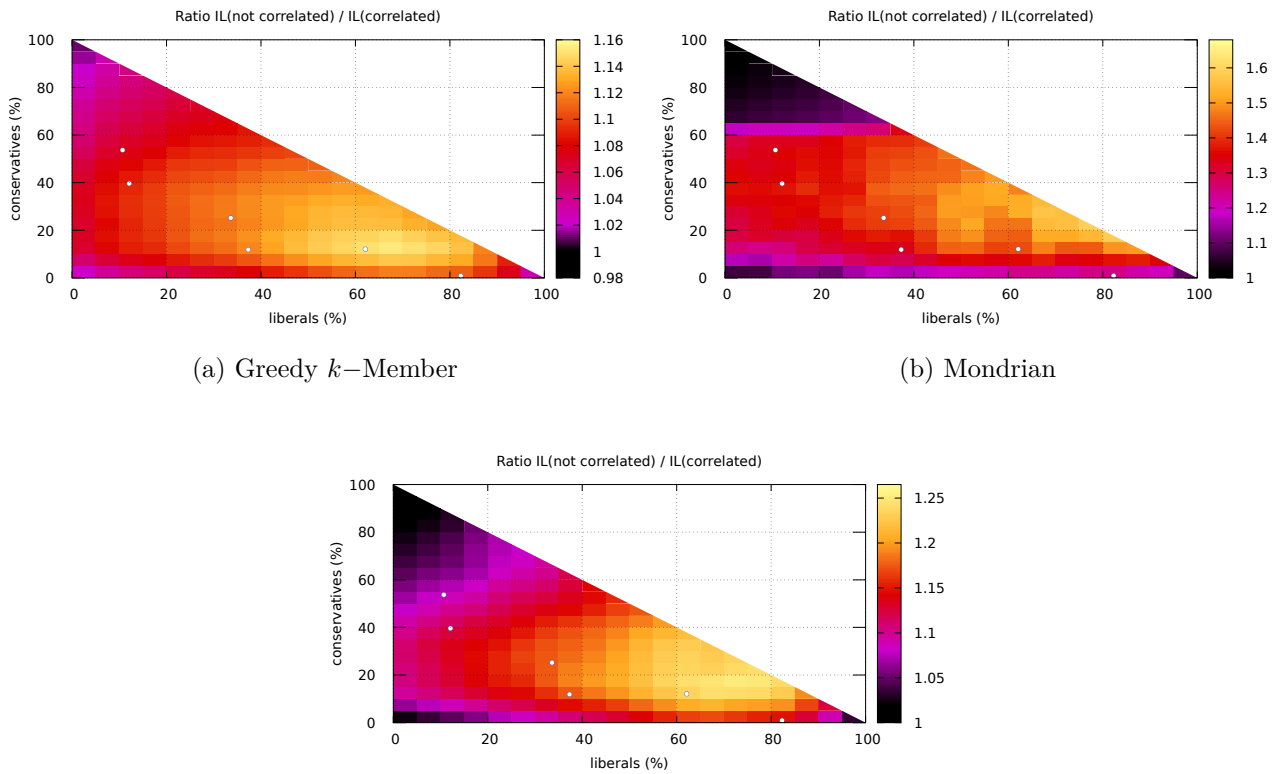


(a) General privacy concern



(b) Data about personal profile

Figure 5.11: Privacy concern distribution over age and education

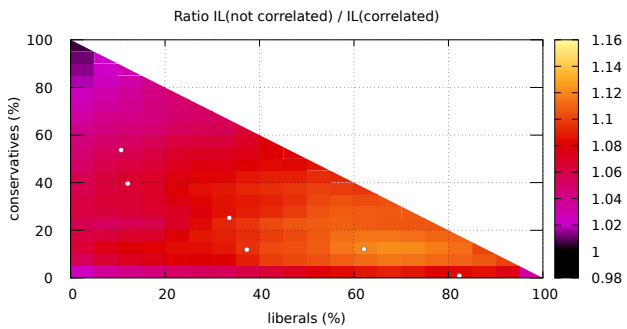


(a) Greedy $k-$Member

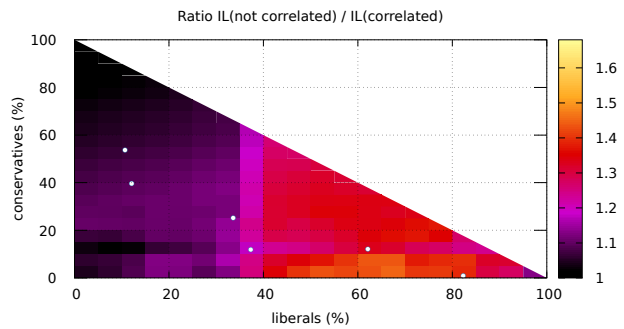
(b) Mondrian

(c) MDAV

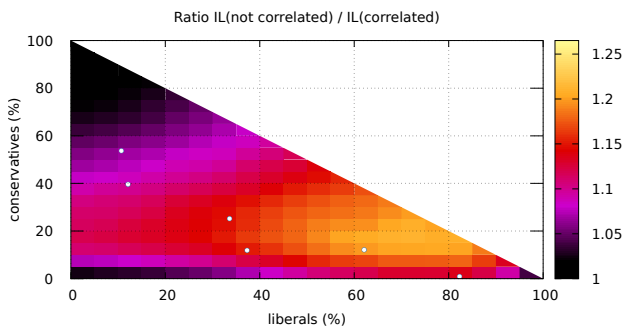
Figure 5.12: Comparison of k_i-anonymity withoout correlation and k_i-anonymity with correletaion by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 3$, $k_m = 5$ and $k_c = 7$



(a) Greedy $k-$Member



(b) Mondrian



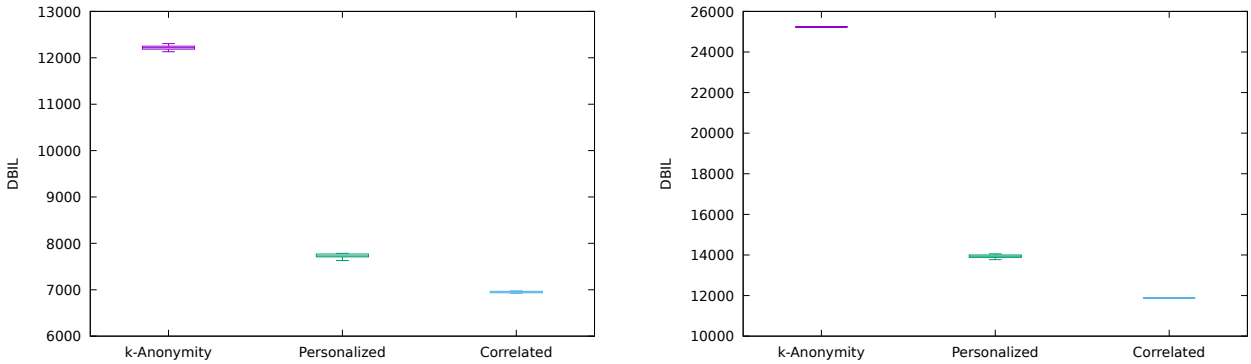
(c) MDAV

Figure 5.13: Comparison of k_i-anonymity without correlation and k_i-anonymity with correlation by varying f_l and f_c ($f_m = 100 - (f_l + f_c)$) with $k_l = 5$, $k_m = 7$ and $k_c = 10$

see that the impact of correlated privacy constraints is higher when near the *sexual and political identity* distribution and fades when near the corner of the graphs (*i.e.* when conservatives, moderates or liberals frequency is near 100%). It is due to the fact that when the proportion of a single group (*i.e.* conservatives, moderates or liberals) is high, records of this group are already gathered together (other are negligible).

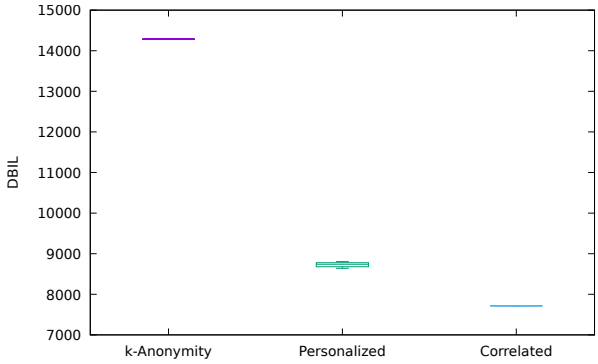
Figure 5.14 shows box plots of the different heuristics. The leftmost box represents the computation of $k-$anonymity using $k = k_c$. The middle one shows the information loss generated by k_i-anonymity when privacy constraints are not correlated to age and education. Finally, the rightmost box represents the computation of k_i-anonymity with correlation. Figures (a), (b) and (c) were computed with the *personal profile* distribution while figures (d), (e) and (f) were computed with the *general privacy concern* distribution. Since only the Greedy $k-$Member heuristic adds randomness to the construction of equivalence classes, it is the only one to have a box when computing $k-$anonymity. The MDAV and Mondrian heuristics are deterministic and so, computing $k-$anonymity with no change in the dataset always gives the same solution (*i.e.* same information loss). The privacy constraints are chosen deterministically when correlated to age and education attributes and suffer from the same effect. Those figures are consistent with section 5.5.2 since the *general privacy concern* distribution shows almost no improvement on data quality when data are not correlated while the *personal profile* distribution shows a great improvement. Also the correlation of privacy concern and data always shows improvement in term of data quality.

The previous figures 5.3 and 5.4 showing the original heuristics together with their adaptation while varying the size of the dataset also show a curve labeled «*Correlated*» which illustrates the impact of correlation on the modification of the size of the dataset. The density of a dataset impacts the information loss generated by the computation of $k-$anonymity. Since the attributes domains do not change, increasing the number of record permits to increase the number of similar neighbours of records. This has the effect to reduce the information loss when passing a threshold (which explains the fact that the graph is a concave curve). This threshold is lower when dealing with correlated constraints. This is even more visible with the Mondrian heuristic which generates lower information loss with 30000 records than with 25000 when constraints are high.

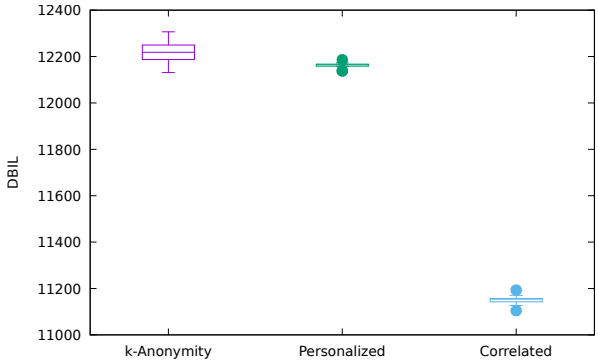


(a) Greedy $k-$Member

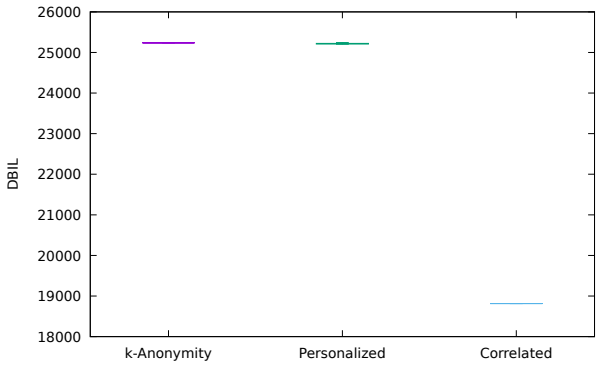
(b) Mondrian



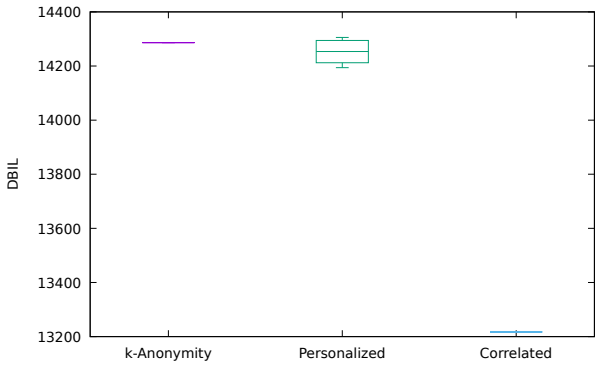
(c) MDAV



(d) Greedy $k-$Member



(e) Mondrian



(f) MDAV

Figure 5.14: Box plot of the k_c-anonymity and k_i-anonymity with and without correlated constraints

Heuristics	Privacy constraints	General privacy concern	Data about personal profile	Data about professional profile	Data about sexual and political identity
Mondrian	Low	1.001 / 1.341	1.813 / 2.124	1.167 / 1.616	1.306 / 1.761
	High	1.000 / 1.064	1.453 / 2.057	1.098 / 1.187	1.065 / 1.427
$k-$Member	Low	1.005 / 1.096	1.581 / 1.758	1.144 / 1.321	1.210 / 1.439
	High	1.001 / 1.064	1.385 / 1.505	1.093 / 1.239	1.112 / 1.302
MDAV	Low	1.002 / 1.081	1.636 / 1.851	1.147 / 1.341	1.214 / 1.489
	High	0.999 / 1.068	1.429 / 1.575	1.104 / 1.274	1.121 / 1.346

Table 5.3: Comparison of k_c-anonymity and k_i-anonymity (top left corner without correlation and bottom right with)

5.5.6 Recapitulation and conclusion

Table 5.3 recapitulates results of the evolution of information loss when varying the distribution of conservatives, moderates and liberals. The corner top-left of table cells shows the ratio information loss generated by $k-$anonymity with $k = k_c$ over information loss generated using k_i-anonymity without correlation. The corner bottom-right of table cells represents the same ratio but with constraints correlated to age and education. The privacy constraints part indicates if constraints were *low* (*i.e.* $k_c = 7$, $k_m = 5$ and $k_l = 3$) or if they were *high* (*i.e.* $k_c = 10$, $k_m = 7$ and $k_l = 5$). The measures given by the table have been made on a sample of distributions specified by A. Acquisti and J. Grossklags and illustrated by table 5.2. As specified by section 5.5.2, distributions of privacy concern play an important role on the information loss generated by the k_i-anonymity. Note also that when conservatives are frequent, the information loss generated by k_i-anonymity is the same as the information loss generated by applying k_c-anonymity as shown by experimentations done using the *general privacy concern* distribution. However, as specified by section 5.5.5, when data is correlated to privacy concern the use of k_i-anonymity always gives an improvement in term of information loss. This effect is due to the fact that a tuple shares the same privacy constraints with others nearby tuples. This leads the three heuristics to make equivalence classes of records with the same privacy contraints leading conservators to not *absob* liberals. The impact of the correlation is also higher for the *sexual and political identity* distribution than other distributions. As specified by the previous sections, the Mondrian heuristics is much more impacted by the correlation of data than other methods. It is an important information since it is the most scalable heuristic.

5.6 Synthesis

In this chapter, we proposed a new anonymisation model, called k_i-anonymity, which allows users to define their own value of k in the context of $k-$anonymity. We have shown, through a detailed experimentation using the classical adult dataset, enriched with a k_i value for each record, that in most cases it is always better to apply k_i-anonymity model than $k-$anonymity. We believe that this work opens an interesting venue of new heuristics and algorithms based on this model.

Chapter 6

Optimal k_i-anonymity with constrained clustering

This chapter presents a new approach to compute optimal k_i-anonymity using constraint clustering. Even if it focuses on k_i-anonymity, the approach can also be used to compute optimal $k-$anonymity.

The first part presents a set of clustering approaches finishing with constrained clustering. Since our new approach uses constraint programming, we remind how it works. We also present the different scenarios that our approach aims to cover.

After that, the $k-$anonymity and k_i-anonymity models are discussed, with regards to a constrained clustering. Since our approach is inspired by an existing clustering approach, we also compare both models to the model described by this existing approach.

The different filtering algorithms are then, presented. They permit to prune inconsistent partitionnings of the dataset to anonymise.

Since constrained programming uses a search tree to find solutions, some strategies must be made to guide the exploration of the search tree. These strategies are shown after the presentation of filtering algorithms.

Finally, experimentations are presented showing the limitations of our approach and which scenarios can be achieved in practice. Since our approach shows limitations, we also give hints on how to improve it and explain different perspectives.

6.1 Introduction

Big Data is a term designating large datasets. With the development of processes such as *Open Data* or *Smart-disclosure*, *Big Data* is becoming increasingly common. However in some domains, data is still rare and produced in small quantities, often because it is expensive to produce. For instance, in paramedical and cosmetic, all products must be tested before marketing. These tests make medical measures on healthy volunteers with medical equipment over various periods of time. Since those tests are costly, the number of volunteers is often quite small. For example, the french project *ADOpTER* (Analyse de Données issues de Tests volontaiRe[1]) deals with datasets varying from dozens to hundreds of records for a given experiment.

Anonymisation methods often focus on the trade off between scalability and data quality. All heuristics are tested on large datasets such as census datasets (*e.g.* adult [59]). When dealing with small datasets, attribute domain definitions are not necessarily smaller than with large datasets. In most of the cases, attributes in small datasets have the same domain as in large datasets but datasets have a much lower density. The information loss generated by anonymisation methods is not of the same magnitude with dense datasets than with sparse datasets. As specified in section 4.3.3, there is a border where statisticians found no utility of generalized data even if the anonymised dataset is not totally generalized. This case happens more often on small datasets due to the higher impact of generalizations. The necessity to compute an optimal solution for a $k-$anonymity problem is thus higher when dealing with small datasets. Since the computation of optimal $k-$anonymity is an NP-hard problem [72] algorithms computing it are considered to be unfeasible and so current research has focused on the development of heuristic based approaches. Section 3.2.2 presents some algorithms to produce an optimal $k-$anonymised dataset (*e.g. MinGen* algorithm). However, these algorithms do not tackle personalized $k-$anonymity.

This chapter tackles the problem of the computation of optimal $k-$anonymity while taking into account the different privacy constraints of differenr individuals. The idea is to use clustering methods together with constraint programming to provide the possibility to compute an optimal anonymised dataset.

The chapter is structured as follows. First, Section 6.2 provides the ma-

[1]Meaning: data analysis from trials on healthy volunteers

terials required to understand the chapter together with real case scenarios dealing with small datasets. Then, Section 6.3 exposes and compares the $k-$anonymity and k_i-anonymity models. Section 6.4 exhibits the different filtering algorithms used to prune invalid clusterings. Section 6.5 presents different strategies to associate records to equivalence classes. Experimental evaluations are presented in Section 6.6. Finally, Section 6.7 concludes.

6.2 Background and scenarios

6.2.1 Clustering

Clustering problems (*e.g.* in data mining) consist to group similar records from a database into groups called clusters. Clustering techniques have been widely used from statistics to privacy preserving techniques. Clusters can be used to infer new information: detecting correlations between different categories of data, making document classification, etc. For example, in the study of A. Acquisti and J. Grossklags [2], they used the $k-$means algorithm [62] to classify subjects of their study into different groups (*i.e.* conservatives, moderates and liberals). Those groups differ by the privacy concerns they expressed. Various families of clustering techniques are based on different approaches and have different objectives. The most known are presented in this section.

Hierarchical clustering

Hierarchical clustering aims to group records which share good similarity (or have low dissimilarity). Groups are constructed progressively in such a way that some hierarchies are highlighted. The dissimilarity is computed with the help of a metric that specifies the distance between two records (some examples are presented in section 3.4). The distance is useful to compare records with each other but linkage criteria are needed to compare clusters with each others. For example, the single-linkage criterion (see the split metric) compares two clusters with the lowest distance between elements of the two clusters. To compare the distance between two clusters c_A and c_B, the single linkage criterion consists to compare the nearest elements of the two clusters using the following formula 6.1:

$$dist(c_A, c_B) = \min_{i \in C_A, j \in C_B} (dist(i, j)) \tag{6.1}$$

with $dist(x, y)$ the distance between x and y. Conversely, the complete linkage criterion is used to compare opposites elements in clusters (*i.e.* the diameter metric). The following formula 6.2 shows how to compare clusters c_A and c_B using this criterion:

$$dist(c_A, c_B) = \max_{i \in C_A, j \in C_B} (dist(i, j)) \,. \tag{6.2}$$

There exist two kinds of clustering techniques, Agglomerative Nesting methods (*i.e.* AGNES) which begin with any records as a one cardinality cluster and merge similar clusters together (*i.e.* such as Bottom-up methods) and Divisive Analysis methods (*i.e.* DIANA) which begin with all records in a unique cluster then divide this cluster into multiple clusters (*e.g.* such as Top-down methods presented in section 3.2.2). AGNES methods are good to identify small clusters when DIANA are better to identify large clusters.

Centroid-based clustering

Centroid-based clustering is a particular family of clustering algorithms which consists in finding a centroid which correctly represents all the records of a cluster. The well known $k-$means clustering [62] is one of those centroid-based algorithms. It takes random centroids (*i.e.* k centroids with k a given parameter) and associates all records to the nearest centroid. After getting those new clusters, it computes the new centroids of each cluster and reiterates with the new centroid until convergence. New clusters' centroids do not necessarily appear in the dataset. The $k-$mean algorithm aims to minimize the pairwise squared deviations in each clusters (see the WCSS metric in section 3.4).

Density-based clustering

Another clustering family is density-based clustering. As the name suggests, it consists to create clusters by maximizing the density in clusters. A good example of density-based clustering is the DBSCAN algorithm proposed by M. Ester et al. [24]. In statistics, records that are very distant to all others are often considered as noise. They are called *outliers* and can modify the results of clustering algorithms. Since they are distant from others, they

also show a lower density with their neighborhood. Since density based clustering tries to keep a minimal (if not a similar) density in each cluster, they almost always do not take outliers into account. The creation of a cluster by the DBSCAN algorithm works as follows: it takes any record r from the dataset, if r has less than n neighbors in a distance d, n and d being given as DBSCAN parameters, it is considered as an outlier. Else, a cluster beginning by r is built. Each neighbor of the cluster is added to the cluster if it also possesses at least n neighbors within distance d. The cluster grows by iterating on new cluster's neighbors and the creation of the cluster ends when no cluster's neighbor contains n records in a radius of d. When the creation of the cluster is finished, DBSCAN tries to create another cluster with the remaining records. DBSCAN stops when no other cluster can be created.

6.2.2 Constrained clustering

Constrained clustering is a concept which allows the specification of constraints. Those constraints must necessarily be satisfied by the clustering algorithm when giving a partitioning of the dataset. Constraints can be specified to give hints to the algorithm to converge quickly or to improve the quality of the solution. Since personalised $k-$anonymity (and also $k-$anonymity) can be considered as a constrained clustering problem it is important to study the different methods to apply constraints while optimizing a given criterion.

Related work on constraint clustering

Integrating constraints in clustering methods has been studied by many researchers. For example M. Bilenko et al. proposed to add *must-link* constraints and *cannot-link* constraints to the $k-$means algorithm. They express those constraints on couples of records (x, y). The *must-link* constraint consists to force x and y records to be in the same cluster. On the opposite, the *cannot-link* constraint forces x and y records to be in distinct clusters. They define a weight matrix for each constraint which represents the penalty of a violated constraint. The penalty associated to a constraint violation may be inconsistent with the distance between the couple of records considered by the constraint violation. To take this into consideration, they use a learning metric which is computed to represent the distance between records together with constraint violation penalties. A learning metric is a metric

evolving during the algorithm (at each iteration of $k-$means in their work). They apply the $k-$means algorithm while maintaining the learning metric in a way to relax their criterion (*i.e.* WCSS criterion with constraint violation penalties). Indeed, their method allows violation of constraints which should not be the case of $k-$anonymity constraints.

Constraints may be diverse, for instance, A. K. H. Tung et al. [88] proposed a way to apply clustering mechanisms on spatial data with obstacle constraints. Using obstacle constraints is an important task when dealing with spatial data. Most of the time those clustering methods are applied on location data in cities which are full of buildings (*i.e.* obstacles). Their idea is to build a *visibility graph* connecting nodes together when they are pairwise visible (*i.e.* no obstacle between records represented by the nodes). Then the distance between two records a and b is represented by the lowest sum of distances between nodes in the path from a to b in the visibility graph.

To solve constraints, there exit a vast amount of methods. S. Gilpin and I. Davidson have proposed to use a SAT solver to make hierarchical clustering under constraints [37] (such as *must-link* or *cannot-link*) using AGNES methods. The idea is that AGNES hierarchical clustering merges clusters together to find a good amount of clusters. Those merges and constraints can be seen as Horn clauses (disjunction of literals with at most one literal not negated) when translating a clustering problem to a SAT problem. Horn clauses are efficiently solvable which explains why they tried to use horn clauses only. It may be hard or impossible to translate k_i-anonymity constraints (or others partition-based constraints) into horn clauses.

T.-B.-H. Dao et al. proposed a new approach using constrained programming to optimize clustering criteria [17]. Their approach is to model clustering problems as Constraint Satisfaction Problems (CSPs). It exists a number of adapted solvers to resolve CSPs (*e.g.* Choco[2], Gecode[3] to cite some). The advantage of such a solution is that all constraints can be expressed to lead clustering algorithms to a good solution. For instance, in their approach, T.-B.-H. Dao et al. expressed common constraints such as:

- *must-link* and *cannot-link* constraints
- minimal size of cluster: constrains clusters to have at **least** n elements (for a given n)

[2]Choco: www.choco-solver.org
[3]Gecode: www.gecode.org

- maximal size of cluster: constrains clusters to have at **most** n elements
- density constraint: constrains elements of a cluster to have at least n neighbors in a radius of m (for given n and m)

They also use constraints to obtain the optimal optimization of their criteria. T.B.H. Dao et al. showed that various constraints were easy to specify in constraint programming, which is one of the main advantages of this approach.

Considering their work, we thought it may be a good approach to calculate optimal k_i-anonymity and thus, this chapter proposes an adaption of the work of T.-B.-H. Dao et al. to achieve optimal k_i-anonymity.

Constraint programming

Constraint programming is a paradigm where constraints are specified and a solver proposes solutions which satisfy these constraints. It is a form of declarative programming which aims to solve *Constraint Satisfaction Problems.*

Definition 9 (Constraint Satisfaction Problem (*i.e.* CSP))**.** A Constraint Satisfaction Problem is a triplet (X, D, C) where

- X is a set of n variables $X = \{X_1, \ldots, X_n\}$.
- D is a set of n domains $D = \{D_1, \ldots, D_n\}$ such as $X_i \in D_i$.
- C is a set of constraints where each C_i applies a constraint on a subset $X' \subseteq X$.

An instance of a CSP solution is an assignment of variables in X to a value in their respective domain in D which satisfies all constraints in C.

There exists a variation of CSP, *Constraint Optimization Problem* (*i.e.* COP). It consists of a CSP which has an objective function to optimize. With a COP, the solver will find the solution which optimizes the objective function. It is used to find optimal solutions and works as follows: when the solver finds a solution, it computes the objective function and posts a new constraint which asks for a better result for the objective function. The solver then tries to find a solution which satisfies this constraint and when it no longer can, the last solution found is the optimal one. Solvers

use a branch-and-bound algorithm to explore different values of variables X_i. Since branch-and-bound algorithms are of exponential complexity and optimal clustering is an NP-Hard problem (except the optimization of single-linkage criterion, *i.e.* split metric), using a simple branch-and-bound is not feasible even for small scale examples. That is why solvers implement constraints in the form of *propagations*.

A **propagation** is a filtering algorithm which removes impossible values from variables' domain $dom(X_i)$. It consists in pruning branches of the search tree. For example, let us assume that constraint C consists of say $X_1 < X_2$, the propagation would reduce the upper bound of D_1 when the upper bound of D_2 is reduced. By filtering values, propagations offer the certainty that inconsistent branches are pruned from the search tree and impossible values are not taken into account. When a branch leads to an impossible solution, a propagation (the one associated to the constraint not satisfied) will change a domain D_i from D to empty set, $D_i = \emptyset$. When a domain is set to $\emptyset$, a variable cannot have any value and the solver detects that the explored branch is inconsistent. The solver then uses backtracking methods to go back in the search tree. Since propagations are filtering algorithms, it is possible that no inconsistent value is found. That is why the solver uses a branch-and-bound algorithm to explore branches. The branch-and-bound updates the domain of a variable following a given *branching strategy*.

A **branching strategy** is a description of how the solver will choose which variable's domain to change to continue the exploration of the search tree. When no propagation can find a value to prune from the tested solution, the solver needs to make a choice because the actual solution does not satisfy the constraints. This state is a node in the search tree and is called to be *stable*. First, the solver needs to select a set of variable's domains to modify. Then, it must select the way this set will be updated (*i.e.* how to partition the domain). For instance, let a domain D_i be selected for the update. It can be updated by selecting a value to assign to the variable or to reduce the domain in half (or any other partioning). In the first case, the number of choices is equal to $|D_i|$ leading to $|D_i|$ successors from the node while in the second case the number of successors is equal to the number of partitions of the D_i made. Propagations will be applied to a successor to find some solutions and if no solution has been found, the backtracking algorithm will go back to the stable state leaving the possibility to try to find solutions with others successors. This process is reiterated over all stable nodes, generating the search tree.

Benefits from the use of constraint programming are multiple. The first advantage is that declarative programming makes it easy to specify constraints. While different partition-based approaches can be combined (*e.g.* combining $k-$anonymity and $\ell-$diversity with $k \neq \ell$), they generally can be expressed by a constraint. For instance, the $\ell-$diversity and $t-$closeness give constraints on the distribution of sensitive attributes in equivalence classes. Using declarative programming permits to adapt many partition-based approaches letting statisticians specify the shape of the anonymisation they want. The second advantage is that the use of declarative programming permits the separation of the specification of constraints from the way solutions are found. By doing so, it takes benefits from further improvements of solvers and methods to propagate constraints. Our approach brings two contributions:

- The first one is the modelisation of k_i-anonymity into constraint programming. This contribution gives the possibility to resolve k_i-anonymity problems with different solvers taking advantage of their different characteristics.
- The second one is the development of different propagators to ensure k_i-anonymity constraints and the development of new strategies to define the shape of the search tree.

While common anonymisation heuristics try to get as close as possible to the optimal $k-$anonymity, our approach is based on a declaration of the optimal and lets solvers find it.

6.2.3 Scenarios

The next paragraphs present different scenarios where small datasets are used.

INSEE scenario

The INSEE is the french National Institute of Statistics and Economic Studies. This institute collects data, analyses those and disseminates anonymised information about the french economy and society. Some of their datasets are manually collected making datasets of representative samples of the population. Those datasets can have a size of about 1500 records which can be considered as small datasets.

Student evaluation of teaching

In high education, it is a common practice to let students express their feedback about the courses they had [89]. It is an important practice for teachers to improve their courses and the way they teach them. To get a better feedback, it is important to anonymise students to give them trust. Since giving its feedback is optional, the trust students have is important to get a more accurate feedback.

The size of that kind of dataset, can vary a lot. The study of E. C. Kokkelenberg et al. [49] shows the impact of class size on student grades. We can see in their study that classes size varies from 20 to 440 students. Since most of student teaching assessments are grouped by classes, those datasets can vary on the same order.

Social and medical studies

In social and medical studies giving a gratification to subjects is a common practice. By doing so, it is easier to find volunteers participating to the study. However, it may be hard to find a funding adequate to extend subjects sample to a large group. This is why datasets in sociology contains generally a small number of subject. For instance, the study of A. Acquisti and J. Grossklags [2] has been made upon 119 responses on the survey they proposed (participants were given a lump sum of 16$). Medical studies are also made on small samples for different reasons as pointed out by B.K. Nayak [70] (*e.g.* limited ressources, ethic principles). In general, sociological studies with face interviews rarely go beyond a coupls hundred individuals.

Healthy volunteers trials

As presented in the introduction of the chapter, before medical drugs or cosmetic products are marketed, it is important to test their efficiency on individuals. The use of healthy volunteers is a common practice to test those products after they were tested on animal subjects. Those tests always give many constraints to volunteers and may make it hard to find volunteers. For this reason, the participation to these tests is also paid, explaining the small size of the datasets. It is important to note that the size of that kind of dataset can also depend of the country in which the study is done.

Models	[17]	$k-$anonymity	k_i-anonymity
Variables			
$\mathcal{G} = [G_1, \ldots, G_n]$, $Dom(G_i) = [1, c_{max}]$	✓	✓	✓
D (diameter), S (split), W (WCSD)	✓	✓	✓
$\mathfrak{u}$ (number of clusters to create)	✗	✓	✓
I (information loss)	✗	✓	✓
Constraints			
$precede(\mathcal{G}, [1, \ldots, c_{max}])$	✓	✓	✓
$atleast(1, \mathcal{G}, c_{min})$	✓	✗	✗
$\forall i \in [1, n]$, $atleast(k, \mathcal{G}, G_i)$	✗	✓	✗
$\forall i \in [1, n]$, $atleast(k_i, \mathcal{G}, G_i)$	✗	✗	✓
Optimization Criteria			
Diameter criterion: $diameter(\mathcal{G}, D, dist)$	✓	✓	✓
Split criterion: $split(\mathcal{G}, S, dist)$	✓	✓	✓
WCSD criterion: $WCSD(\mathcal{G}, W, dist)$	✓	✓	✓
DBIL criterion: $DBIL(\mathcal{G}, I, dist)$	✗	✓	✓

Table 6.1: Difference between T-B-H Dao et al. model [17], $k-$anonymity model and k_i-anonymity model

6.3 Models

To model $k-$anonymity and k_i-anonymity, we were inspired by the article of T.-B.-H. Dao et al. which presents a methods to use constrained programming to optimize clustering criteria [17]. This section will first present the adaptation of the T.-B.-H. Dao et al. model to $k-$anonymity, highlighting differences then will show modifications needed to achieve k_i-anonymity. Those differences are summarized by table 6.1 alongside the model proposed by T-B-H Dao et al. [17].

6.3.1 $k-$Anonymity model

The model of constrained clustering we are using consists to have a variable for each record representing the ID of the cluster (*i.e.* equivalence class) the record belongs to. Let us consider that the database is composed of n records, $\mathcal{G}$ is the set of variables G_i representing the database's tuples, $|\mathcal{G}| = n$. G_i is the variable associated to the ith record in $\mathcal{D}$ and the value of G_i is the ID of

the cluster it belongs to. Let us consider the maximum number of clusters is c_{max}, the domain of variables G_i is $dom(G_i) = \{1, \ldots, c_{max}\}$. The model uses the work of T.-B.-H. Dao et al. [17] to represent a partitioning of the database into several clusters of minimal size greater than or equal to k. The biggest difference with Dao's model is the fact that the number of clusters in $k-$anonymity is unknown. It is however possible to get an upper bound c_{max} of the number of clusters, $c_{max} \leq \lfloor \frac{n}{k} \rfloor$. Even with an upper bound, the $k-$anonymity problem offers a new challenge since the number of clusters expected is much higher than data mining clustering common objective. As specified in section 3.4, in partitioning algorithms, the DM metric aims to build equivalence classes with low cardinalities but when others metrics are optimized, the number of equivalence classes may not be optimized. Figure 6.1 illustrates the optimization of different metrics. Figure (b) shows the optimization of the DM metric while figure (a) shows the optimization of DBIL metric. Depending on the criterion to optimize, the solver may create clusters with a higher cardinality than k which leads to lesser clusters than the upper bound specified before. Generally, all metrics try to create a maximum number of clusters but some exceptions such as exhibited by the figure happen. The upper bound can be maintained during the search of solutions. We introduce the variable $\mathfrak{u}$ to know how many clusters can be created. Let $\{existing\ cluster\}$ be the set of clusters already created, $|\{existing\ cluster\}| + \mathfrak{u}$ represents the maximum number of clusters. The $\mathfrak{u}$ variable is updated when a record i is assigned to a cluster or the domain $dom(G_i)$ avoids i to be put in a new cluster. While the $\mathfrak{u}$ variable is kept up-to-date, we can post the constraint $G_i \leq \mathfrak{u} + |\{existing\ cluster\}|$ (*i.e.* the new upper bound of the number of clusters). Since the branch-and-bound will create clusters progressively the set $\{existing\ cluster\}$ is empty at the beginning and will contain all clusters at the end (*i.e.* when a solution is found). This set is not a variable because we just need to keep knowledge about its cardinality which is the highest value assigned to a variable in $\mathcal{G}$.

To optimize the different criteria, each criterion needs a variable. So variable D represents the diameter criterion (*i.e.* complete-linkage criterion) value, variable S represents the split criterion (*i.e.* single-linkage criterion) value and variable W represents the *Within-Cluster Sum of Distance* criterion (*i.e.* see WCSD and WCSS metrics) value. We also propose to add the diameter-based information loss criterion (*i.e.* see DBIL metric) which is a more adapted criterion to achieve a $k-$anonymity partitioning while optimizing data quality.

(a) Low cardinality, high information loss

(b) High cardinality, low information loss

Figure 6.1: Creating the most clusters may generate more information loss

Since the solver searches for every solution it does not differentiate mirror solutions. For example, let us say that some records are associated to c_i and c_j, a solution swapping c_i and c_j IDs would be considered as a different solutions while it is actually the same. The *precede* constraint has the effect of avoiding those mirror solutions.

The *atleast* constraint from Dao's model differs from $k-$anonymity because in their model, the number of clusters is something defined. They proposed to give a lower and upper bound on the number of clusters (c_{min}, c_{max}). By doing so, they ensure that at least c_{min} clusters have a non-zero cardinality and no records are associated to a cluster ID above c_{max}. Since $k-$anonymity has no constraints on the number of clusters (even if a solution with more clusters is, in general, a better solution), we removed this constraint. However, we add a cardinality constraint which ensures that all clusters possess at least k records which is also a common constraint in clustering methods.

The model proposed by T.-B.-H. Dao et al. includes three criteria. The diameter criterion aims to minimize the greatest diameter of clusters. This criterion is one of the most used in clustering techniques. Since the greatest diameter is reduced when separating the largest cluster in two, this metric tries to create the highest number of clusters. By doing that, clusters with high density will have more records associated than clusters with a lower density. On the opposite, the split criterion will try to create the lowest number of clusters. For example, let A, B and C be three clusters. The split value will be equal to $min\,[d(A,B), d(B,C), d(A,C)]$ (with $d(A,B)$ the lowest distance between two records from A and B), so merging the two nearest clusters would give a better split value (*i.e.* a greater split value). Since it is conflicting with the objective of $k-$anonymity, we also added a constraint to create the highest number of clusters by ensuring that all solutions will give atleast the same number of clusters than the first solution

found (for the split criterion only). The WCSD criterion is more adapted to $k-$anonymity since it tries to create a high number of clusters while keeping the highest split (*i.e.* maximizing the *between-cluster sum of squares* [91]). The DBIL criterion has been also added to minimize information loss. The DBIL criterion consists to minimize the *diameter-based information loss* as presented on section 3.4 reminded by equation 6.3:

$$DBIL(\mathcal{C}) = \sum_{c \in \mathcal{C}} |c| \cdot diameter(c) \tag{6.3}$$

with $\mathcal{C}$ the set of clusters.

6.3.2 k_i-Anonymity model

The k_i-anonymity model has some differences with the $k-$anonymity model. The biggest difference is that the maximum number of clusters is much more complex to compute. When the $k-$anonymity has a general parameter which permits to compute an upper bound with a constant time, the k_i-anonymity cannot. An upper bound can however be computed in linear time.

Theorem 10 (Upper bound of equivalence classes set cardinality)**.** *Let K be the set of users privacy constraints for k_i-anonymity, $K = \{k_1, \ldots, k_n\}$ and let $\mathcal{E}$ be the set of equivalence classes,*

$$|\mathcal{E}| \leq \left\lfloor \sum_{i=1}^{n} \frac{1}{k_i} \right\rfloor \tag{6.4}$$

The case of obtaining the highest number of equivalence classes would be to have all records i of the dataset in an equivalence class of size k_i but as specified in the previous chapter (see section 5.3), this solution is not necessarily the best in terms of data quality. Instead of having a fixed c_{max} for Dao's model, the k_i-anonymity model uses the $\mathfrak{u}$ variable to fit with the fact that the number of clusters must be given in clustering techniques. The $\mathfrak{u}$ variable is initialised as $Dom(\mathfrak{u}) = \left[0, \sum_{i=1}^{n} \frac{1}{k_i}\right]$ and its upper bound will decrease when going deeper on the search tree because going deeper is equivalent to assigning records to clusters and so decreases the number of remaining clusters to be created.

The *atleast* constraint of the k_i-anonymity model ensures that for any record i in the dataset, the value G_i (*i.e.* the ID of the cluster containing

i) has been associated at least k_i times. This constraint ensures that any solution found by the solver achieves k_i-anonymity.

k_i-anonymity shares the same criteria as the $k-$anonymity since the objectives are the same.

6.4 Propagations

This section presents the propagations (*i.e.* filtering algorithms) being used to delete inconsistent values from the domain's variables which leads to the pruning of the search tree. Those propagators are added together with propagators proposed in [17]. The first propagator consists in computing the maximum number of clusters possible to create (*i.e.* updating the variable $\mathfrak{u}$) as explained in section 6.4.1. Section 6.4.2 presents the propagator destined to determine the possible membership of a record to a cluster. Finally section 6.4.3 shows the propagator which filters solutions generating too high information loss.

6.4.1 Number of clusters propagator

The *Number of clusters* propagator (or *nClusters*) consists to use theorem 10 to update the maximum number of clusters $\mathfrak{u}$ to be created and so the highest ID possible to assign to records' variable. It is illustrated by the algorithm 6.

The propagation works as follows: first it initializes to an empty set ℓ and the minimum number of clusters c_{min} a solution can have (*i.e.* the number of clusters with at least one record). In fact, c_{min} can be considered as a value which is the highest cluster ID assigned in $\mathcal{G}$. Due to the *precede* constraint, when a cluster is created, it gets the lowest available ID (*i.e.* the lowest ID which no cluster already owns in $[1, c_{max}]$). The loop, begining at the fourth line, consists in taking all variables which cannot be used to create a new cluster anymore since the last propagation and to put them in ℓ. When a variable is put in that list, it will not be used to create a new cluster in a deeper node of the search tree (but can in other branches). The loop beginning at the 11th line consists in updating the number of clusters that can be created, *i.e.* $\mathfrak{u}$, and the loop at the 15th line reduces the domain of variables in $\mathcal{G}$. The maximum number of clusters is recomputed by reducing the $\mathfrak{u}$ from k_i-anonymity constraints of records in *ell*. The new maximum number of clusters (*i.e.* upper bound of variables in $\mathcal{G}$) is $\lfloor c_{min} + \mathfrak{u} \rfloor$ and

Algorithm 6 Number of clusters propagator

```
1: procedure nClusters(Set of variables 𝒢, variable 𝔲, set of values 𝒦)
2:     ℓ ← empty set
3:     c_min ← |{created clusters}|
4:     for i ∈ {i | G_i just updated} do
5:         if max(Dom(G_i)) > c_min before update and
6:            max(Dom(G_i)) ≤ c_min after update then
7:             ℓ.append(i)
8:         end if
9:     end for
10:
11:    for i ∈ ℓ do
12:        𝔲 ← 𝔲 − 1/k_i
13:    end for
14:    if ⌊𝔲⌋ has changed then
15:        for i from 1 to n do
16:            dom(G_i) ← {v ∈ dom(G_i) | v ≤ ⌊c_min + 𝔲⌋}
17:        end for
18:    end if
19: end procedure
```

is reported to the domain of variables. Only the upper bound of variables domain is updated and this modification (illustrated by the 16th line) is in constant time $bigO(1)$. Since this bound deals with integer only and the $\mathfrak{u}$ variable is a float, this update does not need to be done at every propagation. It is only done when its floor changes.

The loop from line four to nine need a mechanism to keep the upper bound of each variable's domain to identify whether or not, the variable cannot be use to create a new cluster (*i.e.* the maximum value the variable can take is a cluster id already assigned to another variable). The obvious methods would consist to keep copies of variables as propagator local variables. This method would induce a high memory consumption. Fortunately, the loop can be replaced by *advisors*. An advisor, as presented by M. Z. Lagerkvist and C. Schulte [50], is *trigger* applied to a variable. When the variable's domain is updated, the advisor will be called. Since the advisor is link to a propagator it can be use to gather informations and prepare the propagation. Here, all variables have their own advisor which is called when the variable's domain is updated. Initially, no clusters have been created leading all variables to have their domain's upper bound greater than the minimum number of cluster (*i.e.* there is no solution with no cluster). When an advisor is called, it checks if the upper bound of the variable's domain is lower or equal to the highest cluster ID. In that case, the advisor put the variable in the set ℓ and is then disable. In the other case, the advisor does nothing. Disabling the advisor make it to never be called again ensuring that conditions of the fifth and sixth line are true when an advisor is called.

The use of advisors let the loop from the fourth line to the nineth line to be removed from the propagator. The *Number of clusters* propagator complexity is $O(n)$.

6.4.2 Membership propagator

The *Membership* propagator aims to avoid records to be in a cluster with a size lower than the record constraint. It permits the constraint $\forall i \in [1, n]$, $atleast(k_i, \mathcal{G}, G_i)$ to be ensured. The algorithm 7 describes the Membership propagator.

The propagator begins by initializing the size of clusters (represented by c_i) to zero. The loop of the fifth line computes the number of variables a cluster can be associated with for all clusters. It is the highest size the cluster can reach. This loop can be avoided by keeping up-to-date cluster sizes (as a

Algorithm 7 Membership propagator

```
procedure MEMBERSHIP(Set of variables 𝒢, set of values 𝒦)
    for i from 1 to c_max do
        c_i ← 0
    end for
    for i from 1 to n do
        for j from 1 to c_max do
            if j ∈ dom(G_i) then
                c_j ← c_j + 1
            end if
        end for
    end for

    for i from 1 to n do
        for j from 1 to c_max do
            if k_i > c_j then
                dom(G_i) ← {v ∈ dom(G_i) | v ≠ j}
            end if
        end for
    end for
end procedure
```

local variable for instance). Those sizes change when a domain's variable is modified (*i.e.* the cluster ID is removed for a domain). After maximum sizes of clusters are known, for each records i, the propagator removes the cluster ID from $dom(G_i)$ if the highest size the cluster can reach is lower than the k_i-anonymity constraint k_i the user wishes.

6.4.3 Diameter-based information loss propagator

The *diameter-based information loss* propagator consists in optimizing the DBIL criterion. It aims to have the lowest information loss following the DBIL metric (*i.e.* see section 3.4) and to prune branches generating too much information loss.

The DBIL propagation keeps two values up-to-date. The first value is the information loss generated by variables assigned to a cluster. This information loss is computed by applying the DBIL equation 6.3 on clusters with records in it. The second value is an estimation of the minimum information loss records not assigned can generate. Two cases appear: in the first case, the unassigned records can be used to create a new cluster with others unassigned records. In that case, the information loss generated by a record r is the k_r^{th} nearest pairwise distance from r to any others unassigned records. The equation 6.5 shows another form of the DBIL metric:

$$DBIL(\mathcal{D}^*) = \sum_{r \in \mathcal{D}} \cdot diameter(C_{\mathcal{D}^*}(r)) \tag{6.5}$$

with $C_{\mathcal{D}^*}(r)$, the cluster (*i.e.* equivalence class) associated to the record r in the anonymised dataset $\mathcal{D}^*$. In this equation we can see that each record generates a computable information loss which is used to compute a *weight*. The weight of a record is the lowest information loss the record can generate when creating a cluster with it. The second case consists to merge a record to an existing cluster. The information loss generated by this record is the difference of information loss generated by the cluster with and without the record (see equation 6.6):

$$diff_{DBIL}(r, c) = |c \cup r| \cdot diameter(c \cup r) - |c| \cdot diameter(c) \tag{6.6}$$

with $diff_{DBIL}(r, c)$ the difference of information loss generated between c and $c \cup r$. $c \cup r$ is the equivalence class with any record in c plus r. The minimum information loss generated by an unassigned record is the lowest value between the information loss due to the first case and the second case.

Algorithm 8 Diameter-based information lost propagator

```
1: procedure DBIL(Set of variables 𝒢, variable I)
2:     aIL ← 0
3:     uIL ← 0
4:     for i from 1 to c_max do
5:         c_i ← empty set
6:     end for
7:     for i ∈ {i | G_i assigned} do
8:         c_{G_i}.append(i)
9:     end for
10:
11:    for each c_i do
12:        aIL ← aIL + DBIL(c_i)
13:    end for
14:
15:    for i ∈ {i | G_i unassigned} do
16:        d ← ∞
17:        for each c_j do
18:            if diff_DBIL(G_i, c_j) + aIL > I then
19:                dom(G_i) ← {v ∈ dom(G_i) | v ≠ j}
20:            end if
21:            d ← min(d, diff_DBIL(G_i, c))
22:        end for
23:        uIL ← uIL + min(d, k_i^th nearest from {a ≠ i | G_a unassigned})
24:    end for
25:    dom(I) ← {v ∈ dom(I) | v ≥ aIL + uIL}
26: end procedure
```

The filtering algorithm 8 begins with the initialisation of aIL value, the information loss generated by assigned variables, and uIL value, the information loss generated by unassigned variables. Then, clusters c_i are initialised as empty set. The loop from the 7th line consists in filling clusters with records assigned to them. The cluster c_{G_i} corresponds to the cluster associated to the variable G_i (or to the record i). For assigned variable G_i, we consider that G_i is the ID of the cluster in which the variable is assigned. The loop beginning at the 11th line computes the information loss generated by assigned variables by summing information loss of each cluster even if some clusters may not be complete (*i.e.* may have a cardinality lower than k_i-anonymity constraints). The last loop, at the 15th line, computes the information loss of unassigned variables. It also prunes association between clusters and records when it generates a too large amount of information loss. For each cluster, the influence of unassigned records to clusters' information loss is computed using equation 6.6. When the addition $diff_{DBIL}(r, c_i) + aIL$ is greater than the maximum information loss allowed (*i.e.* upper bound of $dom(I)$), the record r cannot be associated to the cluster c_i and i is removed from the domain $dom(G_r)$. This pruning is done by the 19th line. The algorithm uses a variable d to keep the lowest information loss a record can generate when merged to a cluster. Finally, the minimum between d and the k_i^{th} nearest distance from i to others records unassigned is kept to add it to the variable uIL. At the end of this algorithm, we are sure that the information loss generated by all solution from this branch is no less than $aIL + uIL$. Pruning the lower bound of the I variable, permits to detect inconsistent solutions, *i.e.* when a visited branch generates more information loss than a previous solution found.

6.5 Branching strategies

In this section we present the branching strategies we used to build solutions. Section 6.5.1 shows how elements are chosen to initiate the construction of new clusters. Section 6.5.2 indicates how the solver fills clusters with elements when k_i-anonymity constraints are not satified. Finally section 6.5.3 exhibits how the solver switches between the two previous strategies.

6.5.1 Clusters creation

When no propagation can be done and no solutions came out, this branching strategy will take a record (*i.e.* a variable G_i) not assigned to assign it to a cluster. The algorithm 9 reminds this branching algorithm initially proposed by T.B.H. Dao et al. [17].

The algorithm works as follows: first, it initializes a variable v which will be used to identify the record to assign. Then, a set of distances d is initialized. It represents the highest distance from the record v to any clusters (*i.e.* the $d_i = \underset{r \in c_i}{max}(dist(v, r))$). The loop at the 6th line aims to identify which record will be selected to add it to an existing cluster or to an empty cluster following two conditions. The first condition is to take one record amongst records with the lowest domain cardinality (and still unassigned). This is a well-known technique in constraint programming since it focuses on variables which leads to fewer branches (*i.e.* it leads to $|dom(r)|$ branches). The second condition is to begin with the furthest record from all records. It is a strategy proposed by T.F. Gonzales [40]. His idea is to take points that are the furthest from any other points (*i.e.* outliers) to create new clusters. It is also a strategy used by some $k-$anonymity heuristics such as MDAV or $K-$Member Greddy (see section 3.2.2). This loop can be avoided by sorting records by their $totalDist$ (see the equation 6.7) in descending order:

$$totalDist(r, \mathcal{D}) = \sum_{r' \in \mathcal{D}} dist(r, r') \tag{6.7}$$

When the variable to assign v is chosen (v is an outlier with a small domain), the algorithm computes the distance of this record to all clusters by using the highest distance from v to all points of the cluster. The loop at line 15 computes this distance. After that the cluster ID selected by the loop at the 22th line (represented by the value low), is the nearest cluster to v. It is important to note that the second condition of the `if` on the 23th line: $i \in dom(G_v)$ avoids any clusters whose IDs are not in the domain of variable G_v to be selected. Finally line 27 assigns the ID value low to the variable G_v. After that branching, when the backtracking system will come back to the node, the value low will be removed from the domain $dom(G_v)$ and another cluster will be selected for v.

The maximum number of clusters given by theorem 10 is reached when any record r is in a cluster c of cardinality $|c| = k_r$ which is obviously not a realistic case. This branching strategy will always create new clusters

Algorithm 9 Clusters creation branching

```
1: procedure BRANCHCREATE(Set of variables 𝒢)
2:     v ← 1
3:     for i from 1 to c_max do
4:         d_i ← 0
5:     end for
6:     for i ∈ {i | G_i unassigned} do
7:         if |dom(G_i)| < |dom(G_v)| then
8:             v ← i
9:         else if Σ_{j=1}^{n} dist(i, j) > Σ_{j=1}^{n} dist(v, j) and
10:                |dom(G_i)| = |dom(G_v)| then
11:            v ← i
12:        end if
13:    end for
14:
15:    for i ∈ {i | G_i assigned} do
16:        if dist(i, v) > d_{G_i} and G_i ∈ dom(G_v) then
17:            d_{G_i} ← dist(i, v)
18:        end if
19:    end for
20:
21:    low ← 1
22:    for i from 1 to c_max do
23:        if d_i < d_low and i ∈ dom(G_v) then
24:            low ← i
25:        end if
26:    end for
27:    dom(G_v) ← low
28: end procedure
```

even if this leads to inconsistent solutions. Since it does not take into account k_i-anonymity constraints, we need another branching algorithm to fill clusters before creating new ones. However, when no more clusters can be created, this branching offers a good strategy to assign remaining variables to existing clusters.

6.5.2 Filling clusters

This branching strategy aims to identify all clusters with a user constraint not satisfied and then, to add records generating the lowest information loss to those clusters. Algorithm 10 shows the implementation of this branching algorithm. This algorithm needs to first identify which record is in-

Algorithm 10 Filling clusters branching

```
1:  procedure BRANCHFILL(Set of variables 𝒢, set of values 𝒦)
2:      for i from 1 to c_max do
3:          c_i ← empty set
4:      end for
5:      for i ∈ {i | G_i assigned} do
6:          c_{G_i}.append(i)
7:      end for
8:
9:      id ← 1
10:     for i from 1 to c_max do
11:         for e ∈ c_i do
12:             if |c_i| < k_e then
13:                 id ← i
14:             end if
15:         end for
16:     end for
17:
18:     r ← nearest({i | G_i unassigned, id ∈ G_i}, c_id)
19:     dom(G_r) = {id}
20: end procedure
```

side which cluster. The first two loops beginning respectively at the 2nd and 5th lines fill clusters set c_i with assigned variables such that $c_i = \{j \mid G_j \text{ is assigned and } G_j = i\}$. Like the DBIL propagator 6.4.3, those loops

can be avoided if clusters are kept up-to-date during the research process. After that loop, the algorithm searches for a cluster with a user constraint not satisfied. The idea is to try to first complete clusters before creating new ones. To complete a cluster, the algorithm searches for the nearest record r of the cluster and assigns the cluster ID to the record's variable G_r. The research of the nearest record to the cluster only takes into account records with the cluster ID in their domain. When the backtracking comes back to the node, the ID will be removed from $dom(G_r)$.

6.5.3 Switching strategies

Having multiple branching strategies makes the choice of the best strategy to use for the solver difficult. Most of the time, a branching strategy which cannot propose a modification of the domain of a variable will not be used anymore. The idea of the switching strategies algorithm, illustrated by algorithm 11, is to provide a way to have a malleable branching strategy priority list.

Each braching must have a way to specify if they can propose a branching over some variables, we will call it the `possible` procedure. If it cannot, another branching strategy will be chosen. The *create cluster* branching can always assign values to variables which are not assigned. On the other hand, the `possible` procedure of the *filling cluster* branching can assign a value to a variable only when a cluster c has a lower cardinality than the highest privacy constraint of one of its records, *i.e.* $|c| < \max_{r \in c}(k_r)$. The case where a cluster has a lower cardinality than its records privacy constraints and no other variables from $\mathcal{G}$ can be assigned to this cluster leads to inconsistent solutions and is highlighted by the propagators (*e.g.* membership propagator) instead of the branching strategy.

The *switching* strategy algorithm is straightforward. It first tests if any of the procedures from the set of branching strategy $\mathcal{B}$ can be applied. If no branching can be done (*i.e.* $a = \varnothing$), it means that all variables are assigned since the *create cluster* branching can always be done while some variables are not assigned.

Algorithm 11 Switching strategies

```
1:  procedure SWITCHING(Set of branching strategy B)
2:      a ← ∅
3:      for b ∈ B do
4:          if b is possible then
5:              a ← b
6:          end if
7:      end for
8:
9:      if a = ∅ then
10:         No possible branching
11:     else
12:         apply a
13:     end if
14: end procedure
```

6.6 Experimental evaluation and limitations

This section presents experimental evaluations of optimization criteria on the optimal k_i−anonymity. Firstly, the experimental platform is detailed together with the dataset used. Then, the impact of the different criteria over the information loss (*i.e.* using DBIL metric) is shown. Finally the limitations of the constraint programming approach to solve the k_i−anonymity problem is discussed.

6.6.1 Experiment settings

All the experiments were done on a computer with a CPU Intel Core i7-3770. The CPU possesses a frequency of 3.40GHz. Together with the CPU, 16GB of RAM were available. Implementation of the different algorithms were made in C++ using the API of the Gecode [34] solver. The Gecode solver is an open source toolkit destined to the development of constraint systems. In classical branch-and-bound algorithms, the solver copies the state of variables at each use of a branching strategy. By doing so, the backtracking system can go back quickly to stable nodes. Due to this fact, the memory consumption is multiplied by the height of the search tree h. This leads to a memory consumption in $O(n \cdot h)$ with n the number of variables. Since the height

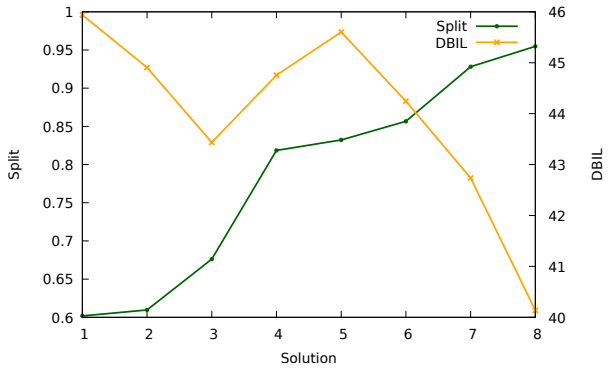


Figure 6.2: Split criterion time line

of the tree can be equal to the number of variables to assign and that every record is a variable, the memory consumption can grow quickly. Gecode offers the possibility to reduce this consumption by not saving all stable states. This has the effect of increasing the computing time but reduces the memory consumption.

First, the optimal $k-$anonymity is a NP-hard problem [72] which shows that it cannot be computed in polynomial time on a random instance. Branch-and-bound algorithms are exponential which induces a high computing time and a high memory consumption. Because of those facts, a random sample of the adult dataset [59] was used to run experiments. The size of sample varied depending on the criterion used. Since the dataset is small, the k_i-anonymity constraints were chosen between two and three, $\forall k_i, 2 \leq k_i \leq 3$. We set liberals to have $k_l = 2$ and moderates to have $k_m = 3$. We chose a distribution of 80% of liberals and 20% of moderates which is near the distribution of personal profile in the paper of A. Acquisti and J. Grossklags (*i.e.* 82.3% of liberals, 16.8% of moderates and 0.9% of conservatives).

6.6.2 Impact of optimized criteria over information loss

This section presents the impact of the different optimization criteria over the DBIL metric. The idea is to see how criteria are useful to find good solutions of k_i-anonymity problems.

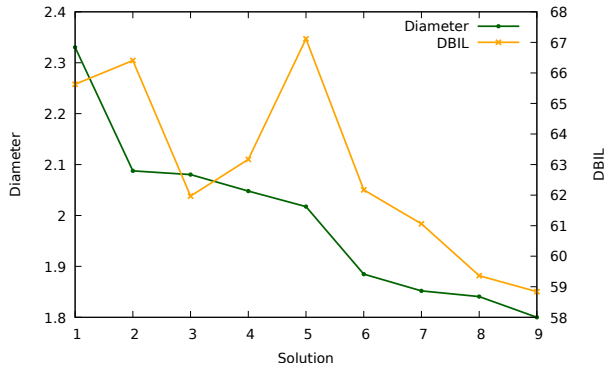


Figure 6.3: Diameter criterion time line

Impact of split function

Firstly, we will present the impact of the optimization of the split criterion over the DBIL metric. Figure 6.2 shows an example of execution of the solver. This figure exhibits the DBIL value alongside the split value of the different solutions the solver has found. The x-axis is the number of the different solutions in chronological order. For example, solution i is the ith solution found which is better than solution $i-1$ and worse than solution $i+1$ in terms of split value. This figure aims to provide a view of the evolution of optimized criterion and of the information loss using the DBIL metric. As we can see even if some solutions found worsen the information loss, the optimization of the split criterion almost always improves the data utility. It is because it aims to create equivalence classes distant from each other which is also a charasteristic targeted by the DBIL metric. Since the split criterion tries to create the lowest number of clusters, we have put a constraint on the minimum number of clusters to create. This leads some solutions to create clusters that can be divided into more clusters to reduce the information loss.

Impact of the diameter function

Secondly, we ran experiments on the optimization of the diameter criterion. Like the split criterion, figure 6.3 shows an instance of execution of the solver using the diameter optimization criterion. This figure shows that when the diameter criterion is improved, the information loss is almost always reduced. However, it happens that the information loss increases while the diameter criterion decreases. The optimal solution regarding the diameter criterion

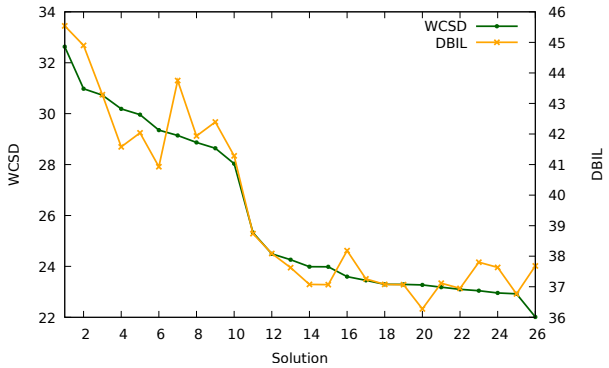


Figure 6.4: WCSD criterion time line

gives an improvement of the information loss. The information loss of the first solution was improved by the optimization of the diameter criterion.

Impact of the WCSD function

The WCSD criterion (*i.e.* generalization of WCSS) is a common criterion in clustering approaches. Figure 6.4 shows the time line of an instance of execution of the algorithm while optimizing the WCSD criterion. We can see that the WCSD criterion is close to the DBIL criterion. It is important to note that computing the k_i-anonymity using the WCSD criterion take more time than computing it using the DBIL criterion but WCSD is also a more studied criterion since the WCSS criterion is used by the most studied clustering algorithm *i.e.* the $k-$means algorithm.

Computation of optimal k_i-anonymity

Finally, we present the result obtained with our new criterion, the DBIL criterion. Figure 6.5 shows the time line of the DBIL criterion. Contrary to other criteria, the DBIL curves is alone on the figure because it is the measure of information loss we use.

Figure 6.6 shows how the optimization of the different criteria improves data utility. The boxes were obtained by computing the ratio of the information loss generated by the first solution found by the solver over the information loss of the optimal solution regarding the criterion. It is important to note that the first solution is an heuristic giving about the same solutions

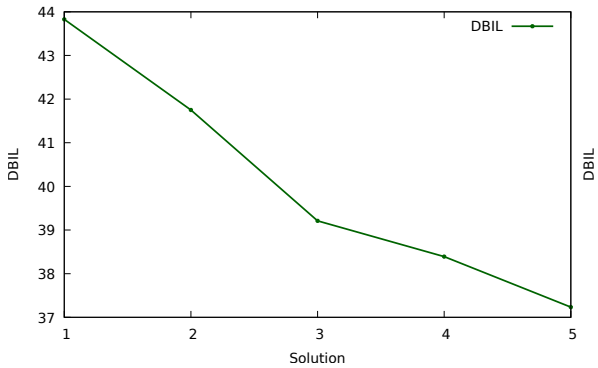


Figure 6.5: DBIL criterion time line

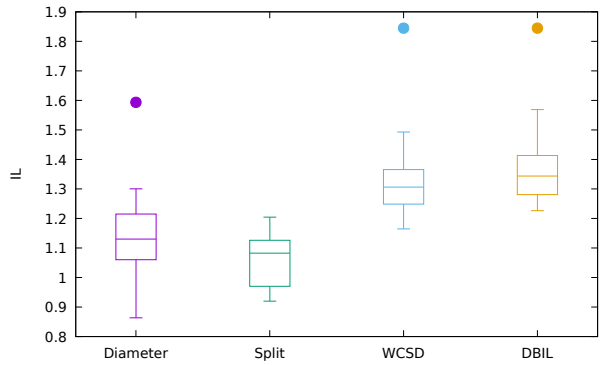


Figure 6.6: DBIL criterion

of the $K-$Member Greedy heuristic. We can see that the use of each of the four criteria gave almost always better solutions. The split criterion is the least efficient. It is due to the fact that the number of clusters created is fixed. Separating big clusters into smaller may improve the solutions given when optimizing the split criterion. On the other hand, the diameter criterion can be used to find better solutions than the split criterion. Since it is the criterion offering the best scalability characteristic over the four criteria, it can be used to compute an anonymised dataset when the optimization of the DBIL criterion cannot. The WCSD criterion gives solutions which are almost optimal regarding the DBIL criterion. Since this criterion is similar to the WCSS criterion used by many clustering algorithms (*e.g.* $k-$means), using those algorithms may give good solutions. Finally the DBIL box shows that the computation of optimal k_i-anonymity can improve the data quality much more than using heuristics. The information loss generated by $k-$member heuristic is generally between 1.3 and 1.4 times the information loss of an optimal solution.

6.6.3 Limitations discussion

While all optimization criteria decrease the information loss, some criteria are more scalable than others. For instance, the diameter criterion can deal with the most records. Our approach can optimize the diameter criterion for a dataset composed of about forty records. On the other hand, the number of records others criteria can deal with does not exceed thirty records. Nevertheless, split criterion could deal with more records than DBIL and WCSD

criteria. This last one could deal with the lowest of records. It is important to note that in the model proposed by T.B.H. Dao et al. [17], the optimization of the diameter criterion could be used for 5000 records datasets while the split criterion could deal with 2000 records datasets. In their model using WCSD criterion did not permit the computation on larger than a 150 records dataset. The efficiency of their solution was reduced when increasing the number of clusters to create but was more affected by the constraint of cluster cardinality lower bound due to the lack of global constraints as explained in the next paragraph. While we do not hope to reach the same size of datasets with our approach, we may reach larger datasets if some improvement are done. Optimal k_i-anonymity can be obtained by almost all scenarios presented in section 6.2.3. However, optimal k_i-anonymity cannot be reached with a small dataset of the INSEE (1500 records). However, it is possible to apply pre-clustering on the dataset (*i.e.* reducing the number of clusters to create). Then apply our clustering methods on clusters created by the pre-clustering. Pre-clustering methods are a common practice also used by algorithms such as the COD-CLARANS[88] or BIRCH [99] algorithms. These methods may be used to handle the INSEE dataset to find an anonymisation which will be locally optimal. The social studies scenario is much closer to accomodate the limitations of our approach. In this scenario, a pre-clustering computation would have a lower impact than with a dataset from the INSEE (*i.e.* which may need multiple pre-clustering). However, datasets from student/teaching evaluations and from healthy volunteers trials would fit perfectly with the computation of optimal k_i-anonymity.

One reason of the limits of our approach is the lack of global constraints. In constraint programming, some constraints may enter in conflict which impacts the different propagations applied by the solver. A global constraint aims to capture simultaneously multiple constraints. The use of global constraints can help models to find solutions quickly [9]. To show the advantage of global constraint let us instantiate a CSP as example:

- Variables: $X = \{x_1, x_2, x_3\}$

- Domains: $D = \{D_1, D_2, D_3\}$ such that $D_1 = D_2 = D_3$ and $|D_i| = 2$

- Constraints: $C = \{C_1, C_2, C_3\}$ such that:

$$\begin{aligned} C_1 =& x_1 \neq x_2 \\ C_2 =& x_2 \neq x_3 \\ C_3 =& x_1 \neq x_3 \end{aligned}$$

This CSP cannot be solved by the propagation of the constraints C_1, C_2 and C_3 separately. It is pretty obvious that this CSP has no solution. However, the solver would need to use its backtracking system to find that no solution exists for this CSP. In contrast, the $AllDifference$ constraint that takes a list of variables and ensures that all those variables are not equal can exhibit that inconsistency. The $AllDifference$ is called a global constraint because it ensures multiple constraints simultaneously. Many other global constraints exist to improve constraint programming methods. For example, C. Bessiere et al. [12] have built a global constraint which ensures the $Precede$ constraint together with the $AllDifference$ constraint. Another example is the use of constraint programming to extract frequent patterns in data mining. In their paper, N. Lazaar et al. [52] uses a global constraint to ensure multiple properties. They show that their global constraint prunes a large amount of inconsistent values in the domain of variables.

6.7 Conclusion

The anonymisation of small datasets is way far from being satisfactory. Most anonymisation approaches deal with large datasets tackling only the *Big Data* problem. The reality is that many use cases do not provide datasets with sufficient data to be classify as *Big Data*. Those datasets can vary from dozens to thousands of records. The computation of optimal k_i-anonymity being possible with those sizes, this chapter proposes an approach to produce an optimal anonymised dataset while adding *user empowerment*. This approach makes use of the constrained programming paradigm which offers many perspectives. The first one is that it can use the technological advances of solvers resolving CSPs. Those solvers are often used in artificial intelligence, a domain which is in broad expansion currently. A second perspective is that the different constraints proposed by partition-based methods can be easily expressed with constrained programming. A third one is that disclosure risk metrics presented by section 3.4.3 can be expressed as constraints to

let the solver find solutions with a low disclosure risk. As privacy is a general concern nowadays, and anonymisation must be done by non-specialists, we believe that a declarative approach is preferable.

Unfortunately, our approach suffers from its lack of scalability making it unefficient on too large datasets. As specified in section 6.6.3, one of the reasons of those limits is the lack of global constraints which is an important improvement in constraint programming. Another reason may also be the estimation of the information loss made by the DBIL propagator presented by section 6.4.3. This estimation is distorted by the fact that it computes an optimal information for each record (which cannot be reached) instead of estimating a global optimal information loss.

Chapter 7

Conclusion (Version Française)

Alors que le nombre de données collectées augmente de jours en jours, les utilisateurs souhaitent reprendre le contrôle sur leur données afin d'améliorer la protection de leur vie privée. Chaque individu devrait être capable de partager ses informations personnelles comme bon lui semble plutôt qu'elles ne soient partagées contre son gré. Ce manuscrit vise à fournir plus de contrôle aux individus sur la façon dont leurs données personnelles sont utilisées. Plus particulièrement, ce manuscrit traite des aspects de personnalisation des protections utilisées pour préserver la vie privée des individus. Ainsi, chaque utilisateur peut exprimer sa propre vision du risque et en déduire le niveau de protection qu'il souhaite.

Ce chapitre offre un bilan des différentes contributions proposées dans ce manuscrit, leurs limitations et les perspectives de recherche qu'elles entraînent.

7.1 Contributions

Cette thèse a présenté un nouveau concept d'anonymisation permettant aux individus d'exprimer à quel point leurs données doivent être anonymisées. Ce nouveau concept, appelé k_i-*anonymat*, a été étudié sur différents scénarios. Ce concept est inspiré du $k-$anonymat. Plutôt que d'utiliser des méthodes telles que la confidentialité différentielle dont la compréhension est hors de portée pour des individus lambdas, nous avons choisi le $k-$anonymat pour sa simplicité. De plus, il est un des rares concepts à ne pas limiter les calculs effectuables sur des données anonymisées (*e.g.* la confidentialité différentielle

s'applique pour un ensemble d'opérations prédéfinies).

Dans un premier temps, le chapitre 4 repose sur la proposition d'une architecture décentralisée (appelé l'architecture *Trusted Cells*) avec laquelle un enquêteur peut interagir. Cette interaction lui permet d'agréger les données stockées dans cette architecture. Cette architecture se compose de deux parties: les *Trusted Data Server* (TDS ou *Serveur de données de confiance*) possédés par chaque utilisateur et une *Supporting Server Infrastructure* (SSI ou *Infrastructure de serveur de soutien*) en laquelle les utilisateurs n'ont pas confiance (on la considère comme *honnête-mais-curieuse*). Les utilisateurs ont confiance envers les TDS et leurs données personnelles sont stockées dedans. Cela leur permet d'y avoir accès et d'obtenir plus de contrôle sur celles-ci.

Les utilisateurs peuvent soumettre des contraintes d'anonymat à leur TDS. Lorsqu'un enquêteur envoie un requête, il y incorpore des garanties d'anonymat. Les utilisateurs participent à la requête en fournissant leurs données seulement si ces garanties sont plus hautes que leurs contraintes. Le protocole élaboré dans cette thèse et, nommé ^{k_i}*SQL/AA*, permet à la SSI et aux TDS de s'entraider afin de calculer les requêtes tout en empêchant la SSI d'inférer des informations personnelles des utilisateurs. De plus ce protocole assure que les garanties données par l'enquêteur sont bien satisfaites et que seuls les utilisateurs ayant des contraintes plus basses que ces garanties, participent à la requête.

Le protocole ^{k_i}SQL/AA se divise en trois phases : la phase de collecte où les utilisateurs décident de participer ou non en fonction de leurs contraintes et des garanties associées à la requête ; la phase d'agrégation qui calcule le résultat de la requête sans que la SSI puisse inférer des informations personnelles des utilisateurs ; la phase de filtrage qui finalise le résultat de la requête avant son envoi à l'enquêteur assurant entre autre que les garanties sont satisfaites. Le protocole ^{k_i}SQL/AA est inspiré du protocole SQL/AA proposé par Q.-C. To et al. [85].

La phase d'agrégation du protocole SQL/AA possédant une limite sur la taille des données possible à traiter, le chapitre 4 propose deux algorithmes permettant d'outrepasser cette limite. Ces deux algorithmes, le *swap-merge* et le *network-merge* permettent d'agréger deux ensembles de données pour n'en former plus qu'un. Le *swap-merge* consiste à utiliser la flash des TDS pour stocker les données lorsqu'elles ne rentrent pas en RAM. En contrepartie, il augmente la charge de travail du TDS. Le *network-merge* consiste à diviser les deux ensembles de données de façon à réduire la charge de tra-

vail des TDS. Il augmente, cependant, le nombre de transferts de données. Ces deux algorithmes permettent aussi de contrôler le temps d'utilisation des TDS et ne sont donc pas limités par le manque de disponibilité des TDS.

Le chapitre 4 propose aussi un processus de généralisation des données afin qu'un enquêteur obtienne plus de réponses à sa requête. L'enquêteur peut proposer d'augmenter les garanties d'anonymat lorsque les utilisateurs fournissent des données moins précises. Cela permet aux utilisateurs avec de fortes contraintes de participer à la requête avec des données moins précises.

Les expérimentations faites en fin du chapitre montrent que le protocole ^{k_i}SQL/AA est tout à fait capable de traiter de larges jeux de données. Il a été testé sur la table *adult* composée de 30162 n-uplets.

Alors que le chapitre 4 propose un protocole pour évaluer des requêtes dynamiques d'agrégation des données tout en respectant les contraintes des individus, le chapitre 5 propose d'anonymiser un jeu de données en vue de le publier. Ce chapitre présente des adaptations de trois heuristiques au k_i-anonymat. Les heuristiques étaient à l'origine conçues pour le $k-$anonymat. Ces heuristiques ont différentes caractéristiques. La première, *Mondrian*, est capable de traiter de plus larges jeux de données mais génère plus de perte d'information. La deuxième, *Greedy* $k-$*Member*, est moins extensible (*scalable*) mais génère aussi moins de perte d'information. La troisième, *MDAV*, présente une solution intermédiaire.

Ce chapitre étudie l'anonymisation de données sous contraintes personnalisées en fonction de trois scénarios. Le premier consiste à proposer aux utilisateurs d'augmenter ou de réduire leurs contraintes d'anonymat. Le deuxième scénario permet aux utilisateurs de réduire leurs contraintes d'anonymat afin d'obtenir des services. Enfin, dans le troisième scénario, un responsable de traitement collecte des données en assurant un degré d'anonymisation fixe. Les personnes souhaitant de plus hautes garanties, ne participent pas (*i.e.* leurs données sont considérées comme supprimées).

Dans le premier scénario, le chapitre montre que dans la plupart des cas relevés par A. Acquisti et J. Grossklags, la perte d'information est plus importante lorsque l'on utilise le k_i-anonymat plutôt que le $k-$anonymat en prenant le k des modérés. Cependant, il est important de noter que dans ce scénario, l'utilisation du $k-$anonymat ne respecterait pas la volonté de nombreux utilisateurs. Il y a tout de même certains cas, tels que des études sur le profil personnel des utilisateurs, où il est possible de satisfaire tous les individus tout en réduisant la perte d'information.

À l'aide de ces expérimentations, le chapitre 5 montre aussi que dans le

cadre du deuxième scénario, utiliser du k_i-anonymat au lieu d'utiliser du $k-$anonymat permet d'obtenir des données anonymes plus précises. Cependant, lorsqu'un grand nombre d'individus choisissent des contraintes d'anonymat élevées, l'utilisation k_i-anonymat permet un gain de précision moindre.

Dans le troisième scénario, le k_i-anonymat induit toujours un gain de précision des données. La suppression de données ayant un impact important sur la qualité des données, il n'est jamais rentable de supprimer des données associées à de trop fortes contraintes.

Le chapitre 5 étudie aussi l'effet de la corrélation entre le choix des contraintes des individus et leur vécu. Il propose une façon de la simuler et montre, dans le cadre du deuxième scénario, que cette corrélation induit toujours une réduction de la perte d'information (même lorsque les individus avec de fortes contraintes d'anonymat sont nombreux).

Enfin, le chapitre 6 propose une toute nouvelle approche pour calculer une solution k_i-anonyme optimale à l'aide de la programmation par contrainte. Pour utiliser la programmation par contrainte, le chapitre définit tout d'abord le modèle du $k-$anonymat et du k_i-anonymat. Ces modèles sont nécessaires pour pouvoir représenter le problème sous forme de variables et de contraintes.

De plus, ce chapitre propose des algorithmes de filtrage et des stratégies de recherche afin de trouver la solution optimale en terme de perte d'information. Ces algorithmes de filtrage permettent de compter le nombre de classes d'équivalence optimal (nommé *nClusters*), d'assurer que chaque contrainte d'anonymat est respectée (nommé *Membership*) et d'optimiser la métrique $DBIL$. Il présente aussi un nouveau système (nommé *Switching strategies*) permettant de sélectionner la meilleure stratégie à appliquer et une stratégie permettant de remplir les classes d'équivalences ne satisfaisant pas les contraintes d'anonymat par des n-uplets les composant.

Ce chapitre étant basé sur le travail de T.B.H. Dao et al. [17], il propose aussi d'utiliser les critères d'optimisation définis dans leur travail. Le premier critère d'optimisation, le *diamètre*, consiste à minimiser le plus grand diamètre parmi les classes d'équivalences. Une autre métrique, le *split*, consiste à maximiser l'espace entre les classes d'équivalences (*i.e.* maximiser la plus petite distance entre des éléments de deux classes d'équivalence distinct). Le troisième critère d'optimisation, la *somme des distances intra-cluster* (*within-cluster sum of distances* ou WCSD), consiste à minimiser la somme des distances entre les points d'une même classe d'équivalence. Ces critères peuvent être utilisés par un statisticien pour obtenir des solutions

optimales s'il estime qu'ils mesurent mieux la perte d'information.

À l'aide des expérimentations menées, ce chapitre montre que ces critères peuvent aussi être utilisés pour réduire la valeur de la métrique $DBIL$. Par exemple, l'optimisation du critère $WCSD$ donne des résultats proches de l'optimisation de la métrique $DBIL$. Ce critère d'optimisation est, néanmoins, moins extensible (*scalable*) que le critère $DBIL$ (*i.e.* les jeux de données sur lesquels il est applicable sont plus petits). Le *diamètre* est le critère le plus extensible et donne des solutions générant peu de perte d'information (*i.e.* moins que des heuristiques).

7.2 Limitations

Nos contributions ont, cependant, des limitations qui sont détaillées dans cette section. La section 7.3 propose des intuitions et perspectives de recherche pour pallier ces limitations.

Le protocole ^{k_i}SQL/AA possède des limitations. Les méthodes d'anonymisation supervisés sont habituellement utilisées avec des outils guidant le *superviseur* en lui permettant, par exemple, d'identifier les données isolées (*i.e.* ne ressemblant à aucune autre). De plus, le superviseur est souvent capable de calculer des statistiques avant et après anonymisation. Il peut ainsi choisir de modifier certaines données anonymisées pour qu'elles soient plus en accord avec la réalité. Dans le protocole proposé dans ce manuscrit, l'enquêteur ne possède aucun outil pour l'aider à choisir les garanties d'anonymat. Cela peut mener un enquêteur à mal estimer les garanties d'une requête. Si ces garanties sont trop basses, il se peut que personne ne réponde à sa requête alors qu'avec des garanties plus élevées, les contraintes des utilisateurs auraient étaient satisfaites. Au contraire, si les garanties d'anonymat fournies par l'enquêteur sont trop élevées, tous les utilisateurs participeront à la requête mais ces garanties ne pourront pas être satisfaites en phase de filtrage ce qui amènera l'enquêteur à ne recevoir qu'une partie des agrégations. Cette limitation est assez importante mais peut être contournée par l'utilisation d'algorithmes d'anonymisation non supervisés.

Une autre limitation du protocole ^{k_i}SQL/AA est le fait que les généralisations proposées par les requêtes soient *linéaire*. C'est à dire, tous les n-uplets sont généralisés de la même façon. Alors que cela peut être vu comme du recodage global (*global recoding*), il ne propose pas de recodages locaux (*local recoding*). Pour permettre aux utilisateurs de faire du recodage local,

il est nécessaire que les généralisations soient proposées sous la forme d'un treillis plutôt que linéairement. Dans le cadre de requêtes de type *GroupBy*, utiliser un treillis augmenterait le nombre groupes compliquant alors les tâche d'agrégation et de filtrage (*i.e.* le résultat de la phase d'agrégation serait plus volumineux).

L'architecture des *Trusted Cells* possède elle-même des limitations. Un adversaire malicieux qui prendrait le contrôle complet d'un TDS aurait la possibilité de déchiffrer les messages échangés par les TDS. Il découvrirait alors les données personnelles des utilisateurs. Une telle attaque serait complexe à effectuer dû au fait que les TDS sont sécurisés matériellement et donc, elle en serait tout aussi coûteuse. Cependant, une telle attaque engendrerait un haut bénéfice. Cette limitation est peu importante puisque l'impact de cette attaque peut être réduit en faisant des groupes de TDS avec la même clé secrète au sein d'un groupe mais des clés secrètes différentes entre les groupes (plutôt que tous avec la même clé). Dans ce cas là, un TDS pourrait seulement effectuer des calculs sur les données des individus de son groupe. Cela réduirait le nombre de TDS disponibles en phase d'agrégation (*i.e.* la SSI peut piocher dans un groupe plus petit de TDS pour effectuer les calculs).

Les algorithmes *swap-merge* et *network-merge* ont aussi des limitations. Le *swap-merge* est limité par la quantité de mémoire flash que les TDS possèdent. Bien que cette quantité puisse être grande, elle reste limitée. De plus, les accès I/O à cet flash étant coûteux et les TDS peu disponibles, le *swap-merge* ne peut pas être utilisé seul. Le *network-merge* permet de pallier au manque de disponibilité et mémoire RAM des TDS. Cependant, il augmente le nombre de communications réseaux. Ces communications peuvent être coûteuses et, augmentent le temps de la phase d'agrégation. Toutefois, la latence d'une requête n'est pas un critère bloquant, la collecte s'étalant elle-même potentiellement sur plusieurs jours. Le temps de mobilisation d'un TDS est par contre un critère plus contraignant car il pénalise l'individu contribuant au calcul. La limitation du *network-merge* prend de l'importance si on considère que les TDS sont tout le temps disponibles et que la latence des requêtes doit être réduite. Par exemple, dans le cas où les TDS sont associés au compteur intelligent *Linky*, ils sont rarement indisponibles. Dans ce cas, le *swap-merge* devient plus accessible, réduisant alors le nombre de communications. Ces deux algorithmes pallient mutuellement la limitation de l'autre, rendant leur limitation peut importante.

Le concept k_i-anonymat, présenté dans ce manuscrit, est limité par le nombre de conservateurs (*i.e.* les individus indiquant de hautes contraintes

d'anonymat). Comme l'indique le chapitre 5, lorsque les données ne sont pas corrélées avec les caractéristiques des individus (*e.g.* âge, niveau d'étude), les conservateurs ont tendance à absorber les libéraux (*i.e.* les individus avec de faibles contraintes) et les modérés (*i.e.* ceux avec des contraintes moyennes). Ceci est dû en partie au fait que les heuristiques présentées dans ce chapitre n'essaient pas de rassembler les conservateurs entre eux. Cela réduit les potentiels gains de qualité des données induits par l'utilisation du k_i-anonymat. Il est nécessaire d'avoir une métrique évaluant la différence de perte d'information entre l'ajout d'un conservateur à une classe d'équivalence ou son exclusion. Cette métrique servirait alors à guider les heuristiques dans la sélection des membres d'une même classe d'équivalence.

Les études présentées dans la section 3.1 montrent que les contraintes d'anonymat des individus sont corrélées à leurs caractéristiques. Dans le chapitre 5, nous simulons cette corrélation de façon *stricte*. C'est à dire que nous regroupons les individus en trois ensembles distincts et considérons que 100% des individus d'un ensemble sont soit libéraux, soit modérés, soit conservateurs. Par exemple, lorsque ce chapitre utilise la distribution *general privacy concern* aucun individu de moins de 36 ans et n'ayant pas fait d'études supérieures (*i.e.* 12th *grade*) ne demande de garanties d'anonymat élevées. Cette corrélation *stricte* n'illustre probablement pas fidèlement la réalité. De plus, les contraintes d'anonymat n'ont pas été choisies en fonction d'autres corrélations existantes telles que la corrélation avec la nationalité d'un individu ou sa confiance envers le responsable de traitement. Cette limitation est importante puisqu'elle empêche d'évaluer correctement l'impact de la personnalisation des protections des individus. Une intuition pour pallier cette limitation est discutée dans la partie «*Génération de jeux de données*» de la section 7.3.

Enfin, la nouvelle approche basée sur la programmation par contraintes possède aussi des limites. La première limitation de cette approche est présentée par la section 6.6.3 et est le manque d'adaptabilité aux jeux de données plus larges. La taille des jeux de données traitables par cette approche reste faible, limitant son impact pratique à des cas particuliers. Un travail de recherche est donc nécessaire pour augmenter la taille des jeux de données exploitables. Des intuitions sont données par la partie «k_i-*anonymat et programmation par contraintes*» de la section 7.3 pour améliorer cette approche.

Le modèle choisi par l'approche utilisant la programmation par contraintes, utilise un grand nombre de variables et de contraintes causant un coup important en mémoire. Ce coût est amplifié par la hauteur de l'arbre

de recherche parcouru par le solveur. Il devient alors, nécessaire de ne sauvegarder qu'un certains nombre d'états stables, nous obligeant à recalculer les états stables non sauvegardés. Cette limitation reste cependant peu importante puisque bon nombre de solveurs proposent de recalculer efficacement les états stables en sauvegardant uniquement les choix des stratégies de recherche. Ainsi, ces solveurs réappliquent ces choix et puis appliquent les propagations (*i.e.* les propagations ne sont pas appliquées après chaque choix).

7.3 Perspectives

Proposer des solutions de *user empowerment* est de nos jours un moyen de mieux protéger les individus tout en obtenant leur confiance. Comme nous l'avons vu dans ce manuscrit, un moyen de contribuer au concept de *user empowerment* est de fournir la possibilité aux utilisateurs de personnaliser leur protection. Le k_i-anonymat est un concept permettant d'appliquer le $k-$anonymat tout en proposant aux individus de spécifier le k qui leur est attribué. Les travaux sur le k_i-anonymat, présentés dans cette thèse, ouvrent la voie à un certain nombre de recherches futures. Nous détaillons ici ces différentes perspectives.

Personnalisation de concepts d'anonymisation. La principale perspective de recherche induite par ce manuscrit est la personnalisation d'autres concepts d'anonymisation. Les concepts basés sur le partitionnement (*e.g.* $\ell-$diversité, $t-$proximité, ...) anonymisent tous les individus uniformément. Le $k-$anonymat ayant des faiblesses, il est nécessaire d'avoir d'autres concepts d'anonymisation personnalisable afin d'améliorer l'anonymisation personnalisée.

***Trusted Cells* et k_i-anonymat non supervisé**. Le protocole ^{k_i}SQL/AA offre un moyen aux individus de personnaliser leurs protections en utilisant le k_i-anonymat et la $\ell-$diversité personnalisée. Le protocole adopte une approche supervisée et dédiée aux requêtes d'agrégation. Une des perspectives possibles serait de proposer une approche non supervisée permettant de produire un jeu de données anonymisé. Dans leur article [3], T. Allard et al. proposent une approche non supervisée pour rendre un jeu de données $k-$anonyme. Leur approche consiste à laisser la SSI créer les classes d'équivalence sans accéder aux informations sensibles. En combinant leur approche avec les heuristiques présentées par le chapitre 5, il est possible

de générer un jeu de données k_i-anonyme. Cette approche permet néanmoins à la SSI d'inférer les quasi-identifiants des individus. Une perspective serait de proposer un protocole permettant aux TDS de créer ces classes d'équivalence sans que la SSI puisse inférer des informations personnelles et tout en garantissant les contraintes des individus.

***Trusted Cells* et confidentialité différentielle personnalisée.** L'utilisation de la confidentialité différentielle consomme le *privacy budget* (le *budget de respect de la vie privée*). Les TDS sont un bon moyen de garder une vue sur la consommation du *privacy budget* de chaque utilisateur. Deux perspectives s'offrent à nous :

- proposer un mécanisme assurant la confidentialité différentielle en phase de collection. Le bruit ajouté sur les données d'un utilisateur par ce mécanisme se superposerait avec le bruit ajouté sur les données des autres utilisateurs. Il est alors nécessaire d'avoir un mécanisme qui se superpose bien avec lui même. Ú. Erlingsson et al. ont proposé le protocole RAPPOR [23] permettant d'assurer la confidentialité différentielle des utilisateurs lorsqu'ils partagent des informations avec un enquêteur. Cette confidentialité différentielle est assurée du côté de l'utilisateur lors de la phase de collecte. Dans leur travail, Ú. Erlingsson et al. utilisent des *filtres de Bloom*. Un utilisateur retranscrit son information dans un filtre de Bloom qu'il bruite. L'enquêteur reçoit un certain nombre de filtres de Bloom bruités, ce qui lui permet de calculer des agrégations sur cette information. L'avantage de cette solution est qu'elle permet aux utilisateurs d'ajouter le bruit qu'ils désirent sans impacter les données des autres utilisateurs. Cependant, elle n'a jamais été testée avec des contraintes personnalisées.

- proposer un protocole permettant l'utilisation d'un mécanisme assurant la confidentialité différentielle en phase de filtrage. Pour cela, il est nécessaire que les TDS attachent le ϵ choisi par les utilisateurs à leurs données. Ainsi, le TDS effectuant les calculs en phase de filtrage pourrait appliquer un bruit satisfaisant les contraintes de tous les utilisateurs.

Génération de jeux de données. Comme nous l'avons vu dans les limitations, la façon dont la corrélation entre les contraintes d'anonymisation et les informations personnelles des individus a été simulée, peut être éloignée de la réalité. Générer un jeu de données de contrainte d'anonymat est quelque

chose de compliqué qui nécessite non seulement des études sociales mais aussi un moyen de générer des contraintes cohérentes avec les résultats de ces études. Une perspective possible est alors de proposer un *framework* (*Boîte à outils*) pour générer des jeux de données avec des contraintes personnalisées.

k_i**−anonymat et programmation par contraintes.** La section 6.6.3 a présenté le concept de contraintes globales en programmation par contraintes. Ce concept permet de spécifier plusieurs contraintes en une seule. Plutôt que de garantir chaque contrainte avec des algorithmes de filtrage individuels, fournir un algorithme qui garantit un ensemble de contraintes permet d'une part, de réduire le temps de calcul et d'autre part de prendre en compte les contradictions entre les objectifs des contraintes. Par exemple, la contrainte exprimant l'optimisation du critère *diamètre* ne prend pas en compte les contraintes d'anonymat et vise à créer des classes d'équivalences ne satisfaisant pas ces contraintes. L'élaboration de contraintes globales est source de nombreuses recherches dans le domaine de l'intelligence artificielle. Actuellement, le diamètre minimum pris en compte par la contrainte *diamètre* est la plus petite distance entre deux points. Pour qu'elle prennent en compte les contraintes d'anonymat des individus, il faudrait qu'elle considère la plus petite distance entre un individu i et le $k_i^{\text{ème}}$ individu le plus proche.

Chapter 8

Conclusion (English Version)

While the amount of collected data increases days to days, users wish to get back the control over their data and improve their privacy. Individuals should be able to share their personal data as they wish rather than being shared against their will. This manuscript aims to give more control to users over their data. More specifically, this manuscript deals with the personalisation of privacy parameters in privacy preserving methods by individuals. Thus, all users can express their own vision of the risk and the level of anonymisation they wish.

This chapter provides an overview of the different contributions presented in this manuscript, together with their limitations, and perspectives they bring.

8.1 Contributions

In this manuscript, a new anonymisation concept has been presented. It allows individuals to express how much their personal data must be anonymised. This new concept, called k_i-*anonymity*, has been studied following various scenarios. It is based on $k-$anonymity. We chose $k-$anonymity for its simplicity rather than using methods such as differential privacy whose understanding is out of reach for the average individual. Furthermore, it is one of the rare concept which does not limit computations that can be performed on the anonymised dataset (*e.g.* differential privacy is optimized for a set of predefined operations).

Firstly, the chapter 4 rely on a decentralized architecture (called the

Trusted Cells architecture) with which a querier can interact. This interaction allows the querier to aggregate data stored in the architecture. This architecture is made of two parts: a large set of *Trusted Data Servers* (TDSs), each owned by an individual and a *Supporting Server Infrastructure* (SSI) which is untrusted by users (we can consider it as an *honest-but-curious* adversary). Users trust TDSs and their personal data are stored in their own TDS. This allow them to access their data and to have a better control on those. The whole database is composed of the data from all TDSs.

Users can submit anonymity constraints to their TDS. A querier that want to make aggregations on the database, adds anonymity guarantees to his query. Users can participate to the query by sharing their data only if anonymity guarantees are greater or equal to their constraints. The protocol made in this thesis and called $^{k_i}SQL/AA$, allow the SSI and TDSs to work together to computes queries while preventing the SSI to infer users personal informations. Furthermore, this protocol ensures that anonymity guarantees given by the querier are met and that only users with lower constraints participate to the result.

The ^{k_i}SQL/AA protocol is divided in three parts: the collection phase in which users decide to participate or not according to their anonymity constraints and the guarantee given by the querier; the aggregation phase in which TDSs compute the result of the query without letting the SSI infer users personal data; the filtering phase in which the result is finalized and the guarantees are ensured. The ^{k_i}SQL/AA protocol is based on the SQL/AA protocol presented by Q.-C. To et al. [85].

Since the aggregation phase of the SQL/AA protocol is limited on the amount of data it can process, the chapter 4 presents two algorithms made to override this limitation. Those algorithms, the *swap-merge* and the *network-merge*, merge two *sorted* datasets in one while computing aggregations. The *swap-merge* uses the flash memory of TDSs to store the data when the RAM is insufficient. Consequently, it increase the TDSs workload. The *network-merge* consists to divide the two datasets in smaller to decrease the TDSs workload. However, it increases the number of data transfer. These two algorithms also allow to control the workload of TDSs and are thus, not limited by the low availability of TDSs.

The chapter 4 presents also a process of data generalisation to allow the querier to get more data from users. To do this, the querier can give higher guarantees for users giving less accurate personal data. This has the effect to keep users with higher constraints in the computation of the query which will

have a more accurate result, even if some individuals data are less accurate.

Experiments made at the end of the chapter show that the ^{k_i}SQL/AA protocol is able to compute query on large datasets. Indeed, the *adult* dataset, composed of 30162 tuples, has been used for those experiments.

While the chapter 4 proposes a protocol for the computation of dynamic aggregation queries while respecting users anonymity constraints, the chapter 5 presents an approach to publish anonymised dataset. This chapter presents the adaptation of three heuristics to k_i-anonymity. These heuristics were originally made for $k-$anonymity and differ from their characteristics. The first, called *Mondrian*, has the best scalability compared to others but also generated the highest information loss. The second, *Greedy* $k-$*Member*, is less scalable but generated an anonymised datasets with the best utility. The third, *MDAV*, is an intermediate solution.

This chapter studies data anonymisation under personalised constraints. We identified three realistic scenarios. The first scenario consists to let users increase or decrease their anonymity constraints. The second scenario consists to propose to users to decrease their constraints to get advantages or services (*e.g.* decreasing the publicity in mobile application). Finally, in third scenario, a data controller collects individuals personal data giving to users anonymity guarantees. People asking for higher guarantees do not participate (*i.e.* their data are considered as removed).

In the first scenario, the chapter shows that in most cases identified by A. Acquisti and J. Grossklags, more information loss is generated when using the k_i-anonymity instead of the $k-$anonymity when using the security parameter of moderates. However, it is important to note that in this case, the use of $k-$anonymity would not respect the wish of many individuals (*i.e.* all conservators). There are still some cases, such as the personal profile distribution, where the use of k_i-anonymity would generate less information loss.

Experiments in the chapter 5, show that in the second scenario, the k_i-anonymity always generates an anonymised dataset with a better utility or, at worse, the same utility than the use of $k-$anonymity. This worse case happens when almost all individuals wish for the highest anonymity constraints which is not a realistic case.

In the third scenario, the k_i-anonymity always induce a better data utility. Indeed, data removing has a real impact on data quality, so, it is never profitable to delete individuals with high anonymity constraints.

The chapter 5 also studies the impact of the correlation between individu-

als privacy concern and their data (*e.g.* age, income). It presents a simulation of this correlation and shows that, in the second scenario, this correlation decreases the information loss induced by the k_i-anonymity (even when there are many individuals with high constraints).

Finally, the chapter 6 presents a new methods based on constraint programming to compute an optimal k_i-anonymous dataset (*i.e.* generating the lowest information loss). To use constraint programming, the chapter firstly defines $k-$anonymity and k_i-anonymity models. These models are necessary to represent the problem through variables and constraints on these.

Then, the chapter presents filtering algorithms, called *propagators*, and search strategies in order to find the optimal solution. These filtering algorithms allow the solver to determine the number of equivalence classes (called *nClusters*), to ensure that all anonymity constraints are satisfied (called *Membership*) and to minimise the $DBIL$ metric. The chapter also describes a new system (called *Switching strategies*) allowing to select the best strategy to apply in order to minimise the information loss and to complete equivalence classes which do not satisfy anonymity constraints.

The solution presented in the chapter 6 is built upon the work of T.B.H. Dao et al. [17] and allows the use of various optimisation criteria. The *diameter* criterion aims to minimise the diameter of the largest equivalence class. The *split* criterion aims to maximise the distance between equivalence class (*i.e.* to maximise the shortest distance between records from distinct equivalence classes). The *within-cluster sum of dissimilarities* (WCSD) aims to minimise the sum of distances between records from the same equivalence class. Those criteria may be used for statistics if a querier assesses that they are more efficient at evaluating information loss.

The chapter 6 shows, via a set of experiments that those criteria can be used to reduce the value of the $DBIL$ metric. For example, the optimisation of $WCSD$ criterion gives solutions similar to optimal solution regarding the $DBIL$ metric. However this criterion is less scalable than the $DBIL$ criterion. The *diameter* is the most scalable criterion and gives solutions generating low information loss (*i.e.* less than heuristics).

8.2 Limitations

Contributions presented in this manuscript have limitations which are described in this section. The section 8.3 shows hints and perspectives to

overpass these limitations.

The ^{k_i}SQL/AA protocol has limitations. Supervised anonymisation methods are usually used with tools leading the supervisor in the anonymisation process. For example, it can help him to identify outliers (*i.e.* tuple different from any others). Furthermore, the supervisor is often able to compute statistics before and after the anonymisation process. By doing so, he can alter some anonymised data to better fit the reality and manually decrease the information loss. In the protocol proposed in this paper, the querier have not access to tools which would help him decide on anonymity guarantees. This may lead the querier to choose wrong guarantees. If anonymity guarantees are too low, most individuals would not participate to the query, and conversely, too high guarantees would be impossible to satisfy leading to the deletion of data in the filtering phase. This limitation may be important but can be overpass by the use of unsupervised anonymity algorithms.

Another limitation of the ^{k_i}SQL/AA protocol is the *linear* generalisations that the querier can make. For example, all tuples are generalised the same way, which can be seen as *global recoding*. However, it does not allow *local recoding*. To allow it, generalisations must be proposed in the form of lattices. The use of lattices would increase the number of groups in a *GroupBy* query. This would complicate the aggregation and filtering phase (*i.e.* the result of the aggregation phase would be larger).

The *Trusted Cells* architecture has its own limitations. Suppose a malicious adversary took the full control of a TDS. He could decrypt any message shared by TDSs and obtain all individuals personal data. Compromising a TDS would be hard and expensive due to the fact it is a secure hardware. However, it would be an attack with a high benefit. This limitation can be reduced by dividing the set of TDSs in multiple clusters. All TDSs in a same cluster would share the same cryptographic key but TDSs from distinct clusters would have different keys. TDSs would be able to compute aggregations on data from TDSs of the same cluster only. This may decrease the amount of available TDSs to compute these aggregations (*i.e.* the SSI can select TDSs from a smaller set of TDSs).

The *swap-merge* and *network-merge* algorithms have also limitations. Firstly, the *swap-merge* algorithm is limited by the amount of flash memory TDSs have. Although this memory may be large, it remains limited. Secondly, I/O access to this memory are expensive and since TDSs have a low availability, the *swap-merge* algorithm cannot be used alone. The *network-merge* is able to overpass the lack of TDSs availability and RAM memory.

However, it increases the amount of network communications. Those communications may be expensive and increase the time of the aggregation phase. However, the query latency is not considered as a limit since the collection phase can take several days. The *network-merge* limitations are more important in the case TDSs are always available and the query latency must be low. For example, in the case TDSs are embedded in the *Linky*, the french smart electricity meter, they would be rarely unavailable. In this case, the *swap-merge* is more efficient, decreasing the number of network communications. Since those two algorithms mutually compensate their limitations each other, these limitations are not important.

The k_i-anonymity concept presented in this manuscript is limited by the amount of conservators (*i.e.* individuals with high anonymity constraints). In the chapter 5, we see that without considering the correlation between data (*e.g.* age, study level) and privacy concern, conservators absorb liberals and moderates. This is partly due to the fact that heuristics presented in this chapter does not try to group liberals and moderates together. This decreases the potential gain of utility the k_i-anonymity proposes. It becomes, then, necessary to have a metric that measures the difference of information loss between adding a conservators to an equivalence class or its exclusion. This metric would lead heuristics to a better solution.

Studies presented in section 3.1 shows that individual anonymity constraints are correlated to their data. The chapter 5, simulates this correlation in a *straight* way. This means that we consider three groups of similar individuals differing from their constraints. One group contains all conservators, another contains all moderates and, finally, the third group contains all liberals. For example, using the *general privacy concern* distribution no individual under 36 years old with no higher education (*i.e.* 12th *grade*) have high anonymity constraints. This *straight* correlation probably does not accurately reflect the reality. Moreover, anonymity constraints are not chosen in function of others correlations such as the correlation between privacy concern and nationality. This limitation is important because it prevents to correctly evaluate the impact of anonymity constraints personalisation. An hint to overpass this limitation is discussed in the part «*Dataset generation*» in the section 8.3.

Finally, the new approach based on constraint programming also have limits. The first limitation of this approach is presented by the section 8.3 and is its lack of scalability. Indeed it cannot process the anonymisation on big datasets. Since the size of datasets it can handle is small, it can

only be used on particular use cases. More researches are needed to increase the amount of data it can handle. To improve this approach, some hints are provided by the part «k_i-*anonymity and constraint programming*» in section 8.3.

The model based on constraint programming uses many variables and constraints which may induce high memory cost. This cost is magnified by the height of the research tree the solver explores. It becomes necessary to save only a small amount of stable states leading to recalculate non save stable states. Many solvers propose an efficient way to do so by saving only choices made by the branching strategy. After applying the choices, the solver can use propagations (*i.e.* propagations are not apply for each choice but after all choices application) to obtain a new stable state.

8.3 Perspectives

Proposing user empowerment solutions is important to gain trust from users while ensuring their protection. As seen in this manuscript, a way to propose user empowerment is to allow individual to personalise the protection of their privacy. The k_i-anonymity is a concept that applies $k-$anonymity while proposing users to set their k parameter. Works on k_i-anonymity presented in this thesis, induce some research perspectives. This section presents these perspectives.

Personalised anonymity concepts. The main perspective induced by this manuscript is the personalisation of others anonymity concepts. Concepts based on dataset partitioning (*e.g.* $\ell-$diversity, $t-$closeness, ...) are all uniform. The $k-$anonymity has its own weakness that others concepts try to compensate. That why, it is necessary to work on their personalisation.

Unsupervised k_i-anonymity under *Trusted Cells* architecture. The ^{k_i}SQL/AA protocol offers to individuals the possibility to personalised their protections using k_i-anonymity and personalised $\ell-$diversity. This protocol can be seen as a supervised anonymisation method dedicated to aggregation queries. One of the researches perspectives ^{k_i}SQL/AA protocol induces would be to propose an unsupervised anonymisation method. In their article [3], T. Allard et al. have proposed an unsupervised approach to compute a $k-$anonymous dataset on the *Trusted Cells* architecture. Their method consists to let the SSI to build equivalences classes by sharing only quasi-identifiers (*i.e.* sensitive data are encrypted by TDSs). By combining

their approach with $k-$anonymity heuristics presented by the chapter 5, we can generate a k_i-anonymous dataset. This approach allows the SSI to infer all individuals quasi-identifiers. Another perspective would be to propose a protocol where equivalence classes are created by TDSs. By doing so, the SSI would not be able to infer any personal information while ensuring individual anonymity constraints.

Personalised differential privacy under Trusted Cells architecture. The use of differential privacy consumes the *privacy budget*. TDSs are able to save the privacy budget consumption of each individuals. We can discern two major perspectives regarding personalised differential privacy:

- to propose a protocol which ensures differential privacy to users in the collection phase. Individuals would add noise to their data independently of the others. Users noise would superimpose during the computation of the aggregation phase. That is why, it is important to have a differentially private mechanism which superimpose correctly with itself. Ú. Erlingsson et al. have proposed the RAPPOR [23] protocol which ensure differential privacy to users when they send their data to a querier. The differential privacy is ensured on the user side during the collection process. In their work, Ú. Erlingsson et al. used *Bloom filter*. A user translates his data as a Bloom filter and adds noise into it. The querier receives number of noisy Bloom filters which allow him to compute aggregations. The benefit of this method is that it allows users to separately add noise on their data without impacting others users data. However, this method has not been experimented with personalised differential privacy.

- to propose a protocol which use a differentially privacy mechanism in the filtering phase. To do so, it becomes necessary that TDSs attach users ϵ to their data. By doing so, the TDS which computes the filtering phase, can add noise to the result regarding users constraints.

Dataset generation. As said in the limitations section, we have simulated the correlation between users anonymity constraints and their data in a way that may not totally reflect the reality. Generating a dataset containing anonymity constraints is complex and need number of social studies and a mechanism that can generate these data consistently with these studies. A possible perspective is to propose a framework for generating personalised anonymity constraints.

k_i-**anonymity and constraint programming.** The section 6.6.3 has presented the concept of global constraints in constraint programming. This concept allows the specification of multiple constraints in one. Instead of having many filtering algorithms that can be redundant, providing an algorithm that can ensure multiple constraints allows, on one hand, to decrease computation time and to take into account incompatibilities between solutions induced by different constraints. For example, the constraint which optimises the diameter criterion, does not take into account anonymity constraints and try to create too small equivalence classes (*e.g.* isolating outliers alone in their equivalence class). In artificial intelligence community, it is common to put lot of effort to elaborate efficient global constraints. Currently, the diameter constraint computes the lowest diameter as the smallest distance between two tuples. To consider anonymity constraints, the constraint could computes the lowest distance between any tuple i and its k_i^{th} nearest tuple.

Personal publications

The works presented in this thesis have been the subject of the following publications:

- **The Case for Personalized Anonymization of Database Query Results** [67], Axel Michel, Benjamin Nguyen and Philippe Pucheral, in *Communications in Computer and Information Science*, a bookchapter of *Data Management Technologies and Applications*.
- **Managing distributed queries under personalized anonymity constraints** [66], Axel Michel, Benjamin Nguyen and Philippe Pucheral, at *DATA conference* 2017.
- **Exécution de requêtes distribuées sous contraintes d'anonymat** [65], Axel Michel and Benjamin Nguyen, at *forum de la gestion des risques naturels, technologiques et sanitaires Envirorisk* 2016.

Figures of chapter 5

This appendix shows figures from the chapter 5 in larger versions. This aims to make these more readable.

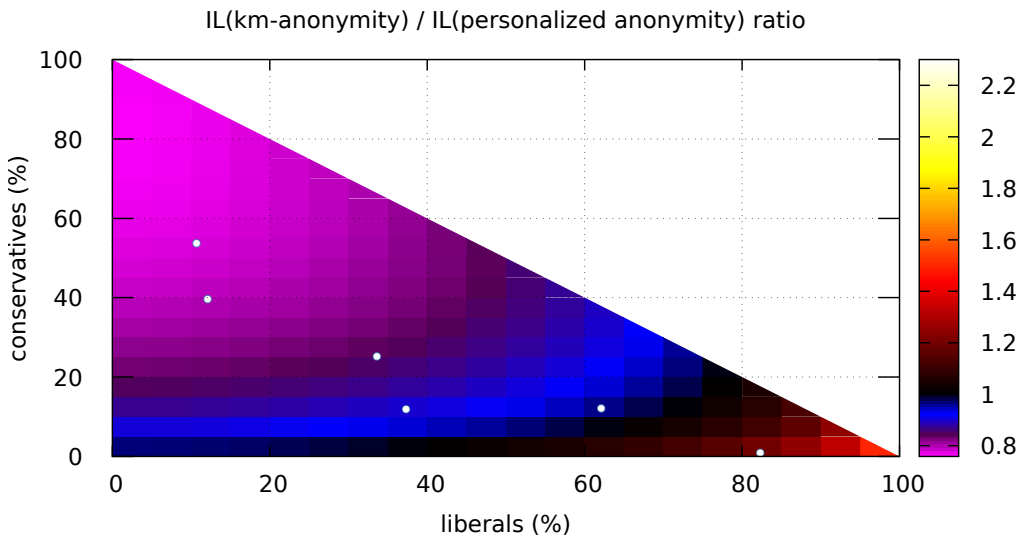


Figure 1: Greedy $k-$Member with low constraints, referring to figure 5.5a

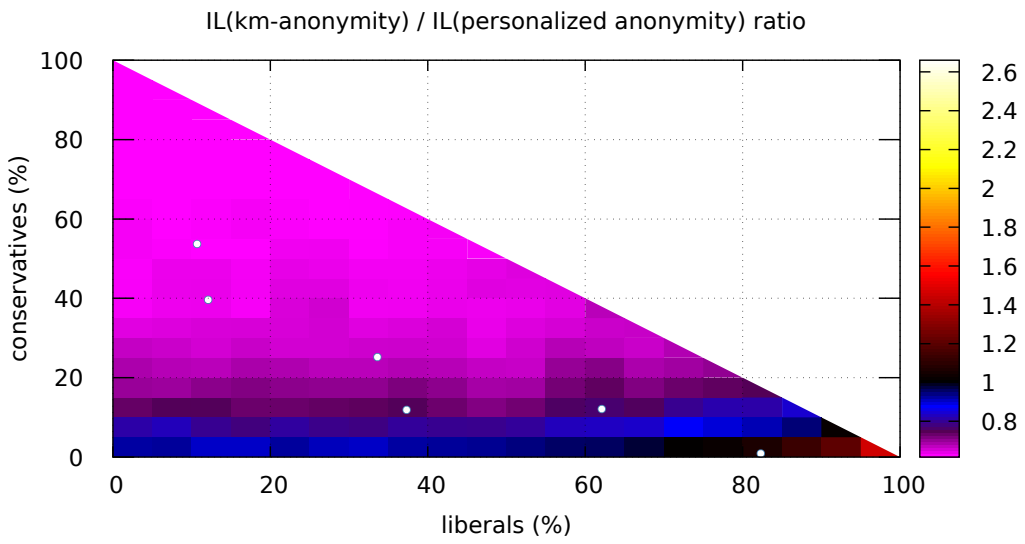


Figure 2: Mondrian with low constraints, referring to figure 5.5b

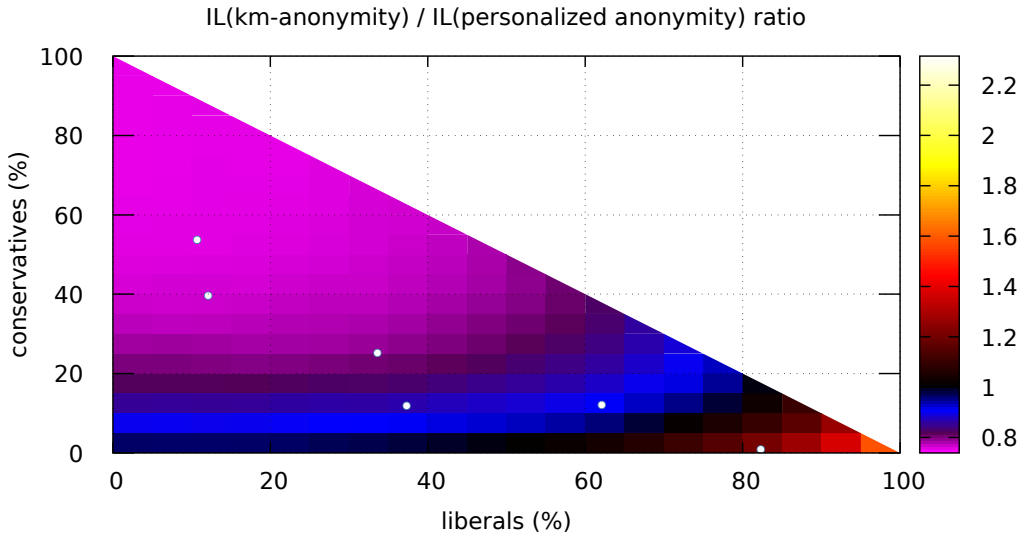


Figure 3: MDAV with low constraints, referring to figure 5.5c

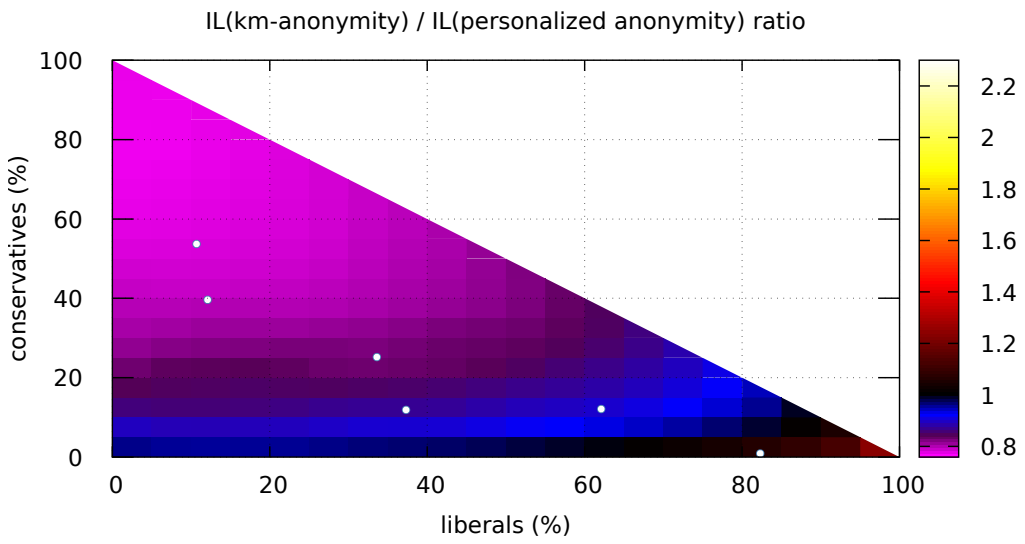


Figure 4: Greedy $k-$Member with high constraints, referring to figure 5.6a

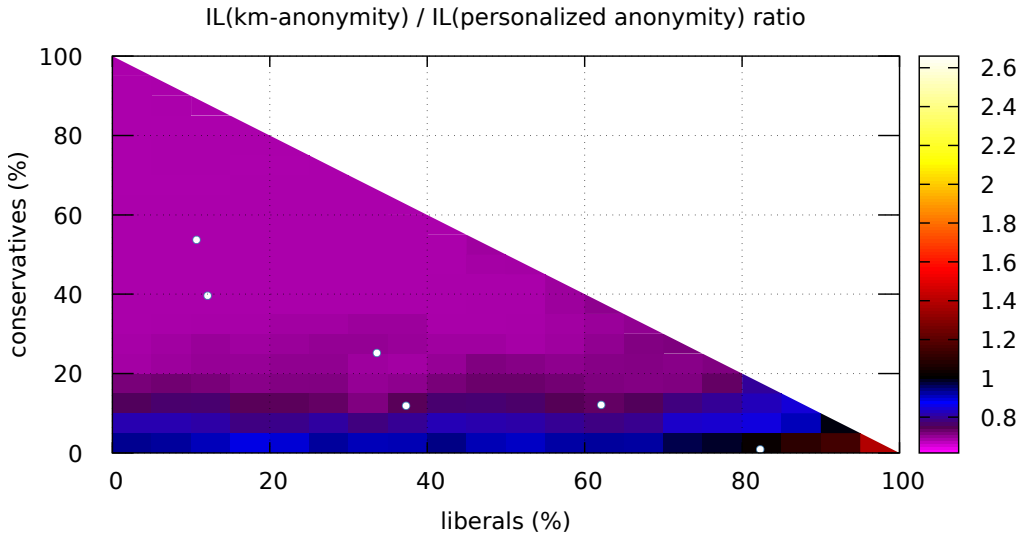


Figure 5: Mondrian with high constraints, referring to figure 5.6b

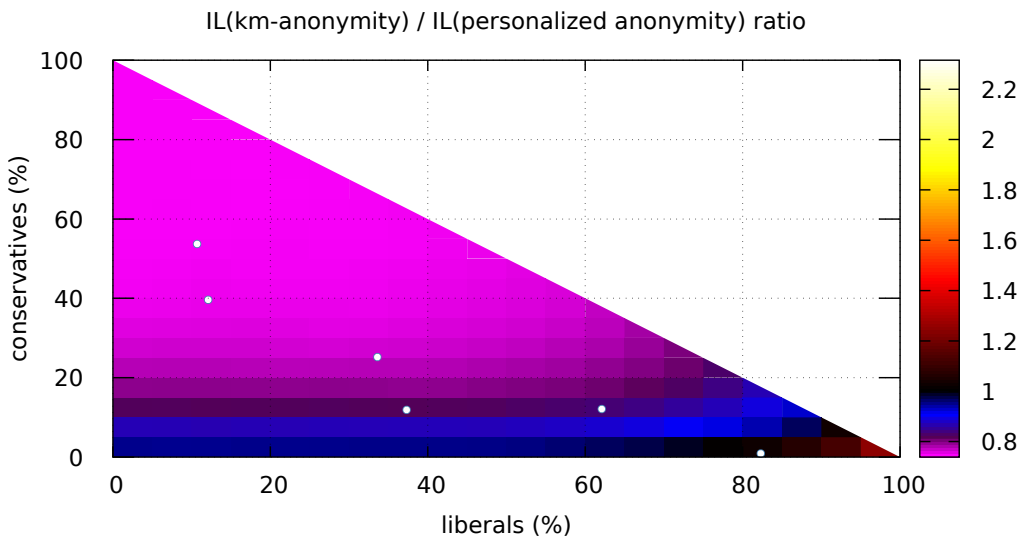


Figure 6: MDAV with high constraints, referring to figure 5.6c

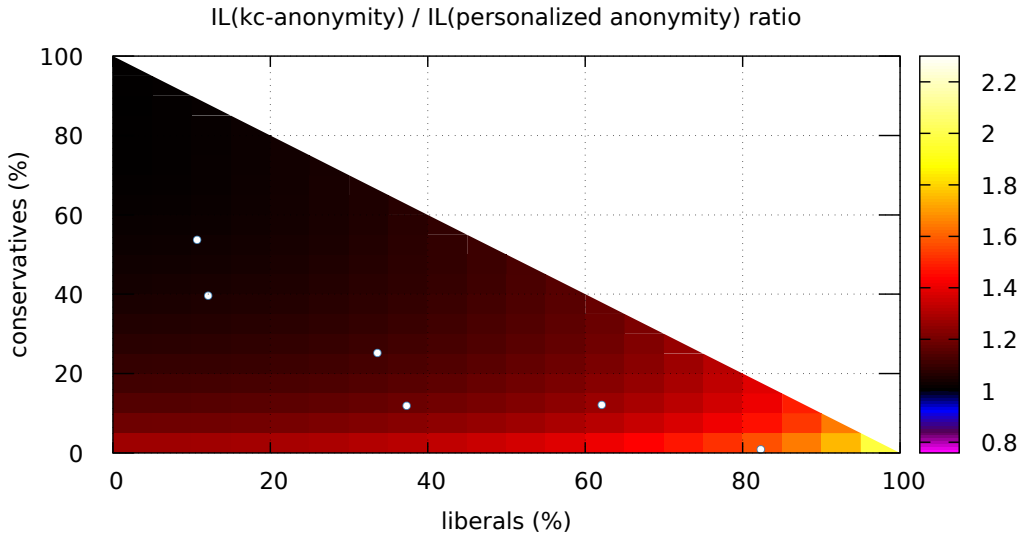


Figure 7: Greedy $k-$Member with low constraints, referring to figure 5.7a

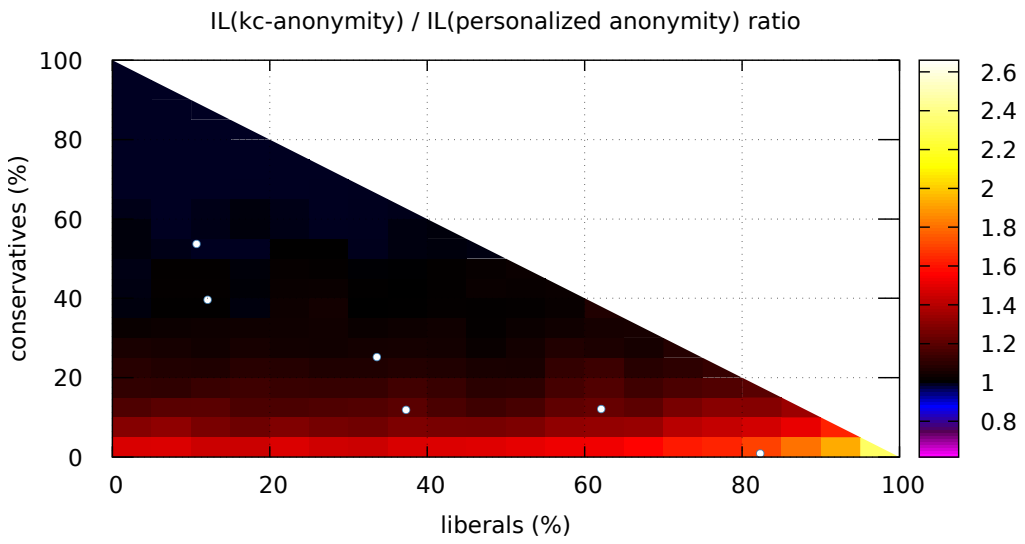


Figure 8: Mondrian with low constraints, referring to figure 5.7b

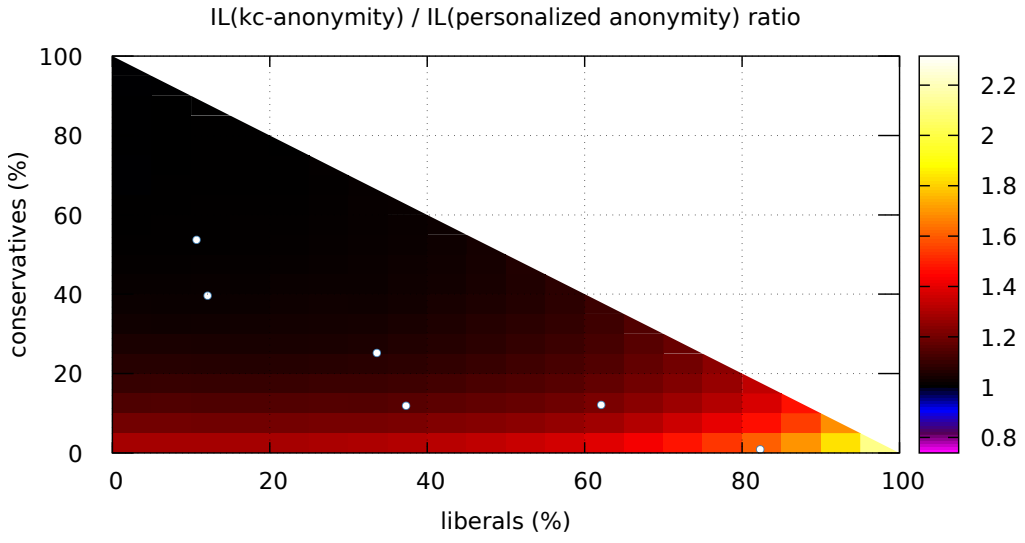


Figure 9: MDAV with low constraints, referring to figure 5.7c

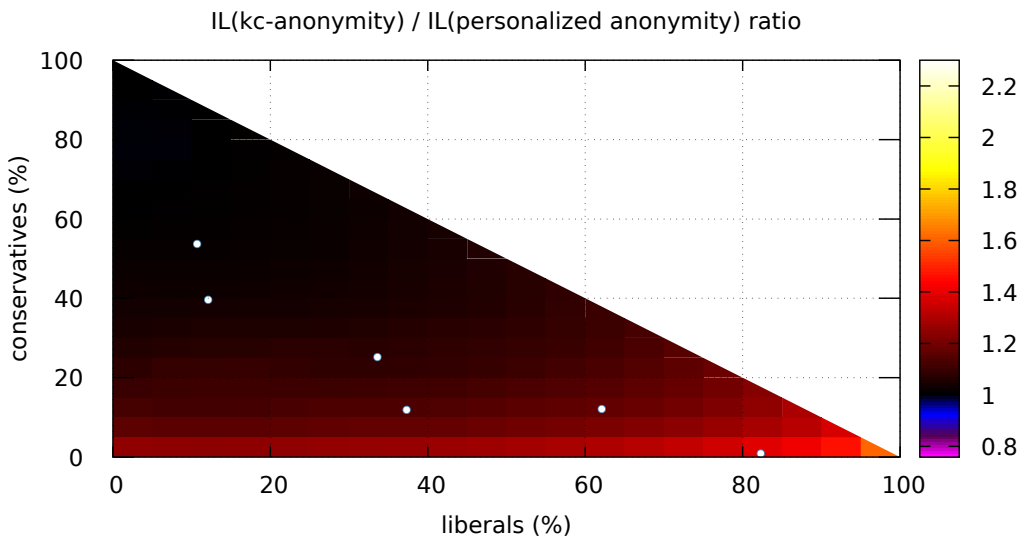


Figure 10: Greedy $k-$Member with high constraints, referring to figure 5.8a

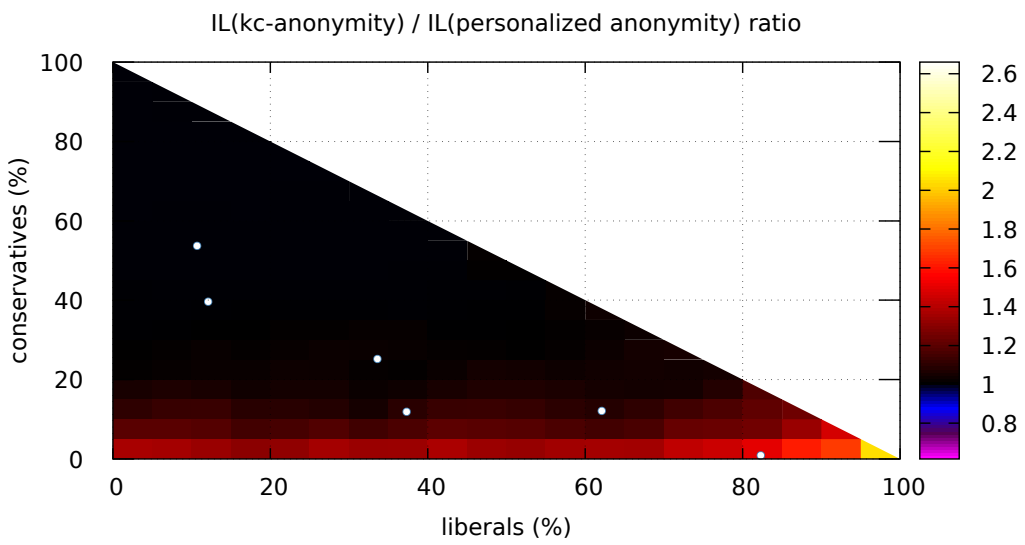


Figure 11: Mondrian with high constraints, referring to figure 5.8b

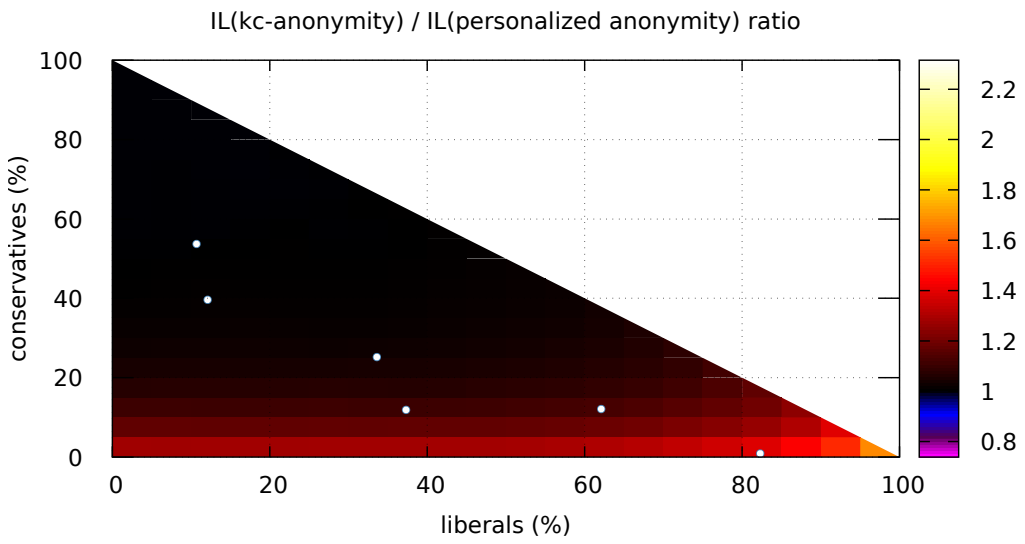


Figure 12: MDAV with high constraints, referring to figure 5.8c

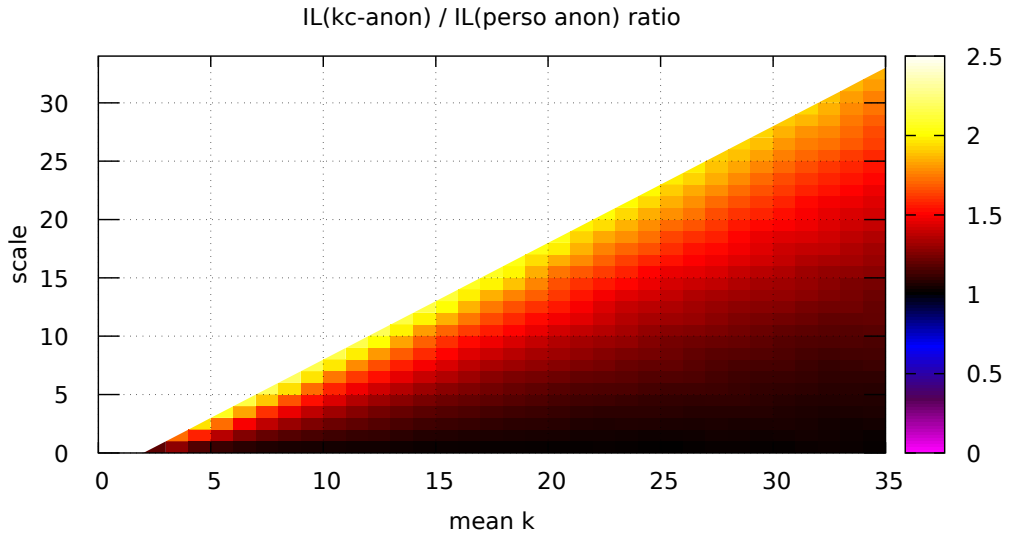


Figure 13: Greedy $k-$Member varying the scale, referring to figure 5.9a

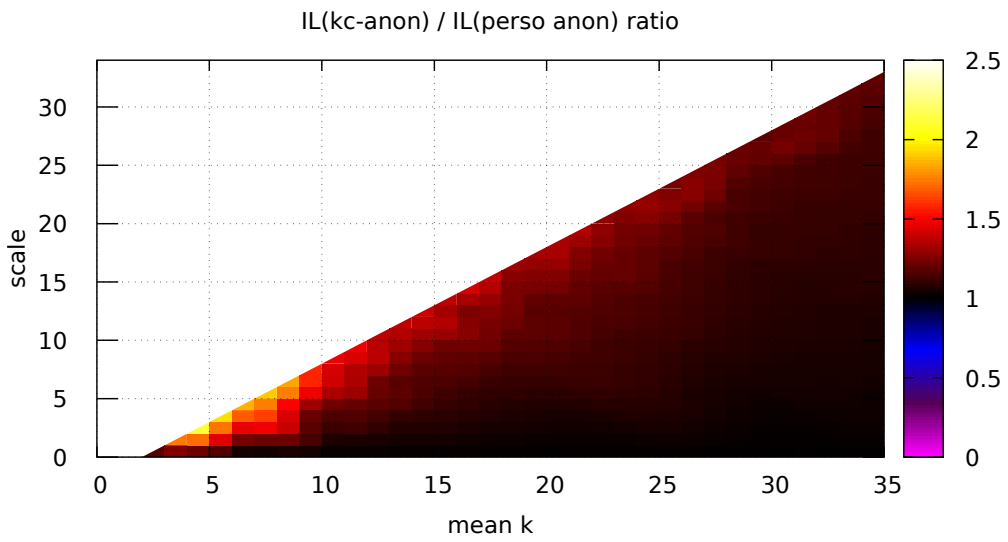


Figure 14: Mondrian varying the scale, referring to figure 5.9b

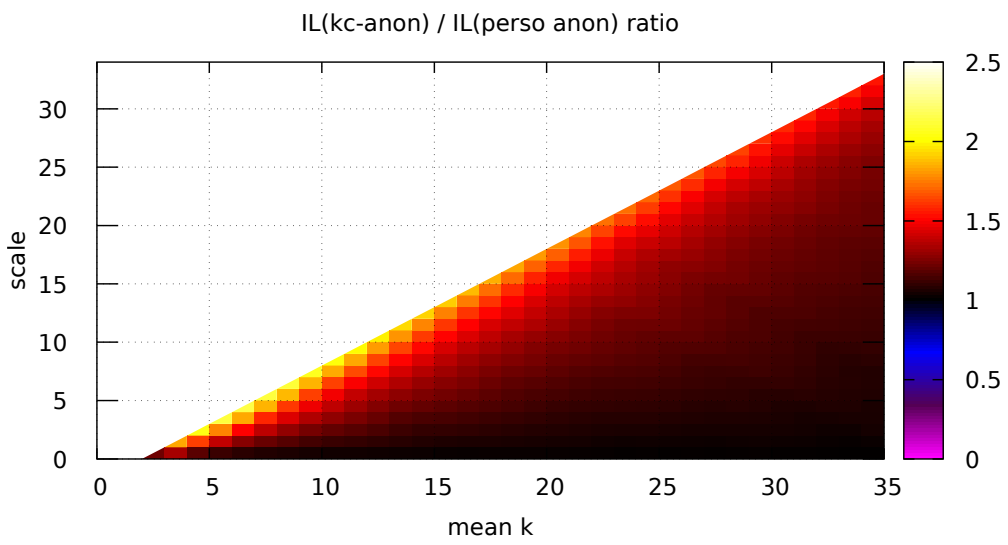


Figure 15: MDAV varying the scale, referring to figure 5.9c

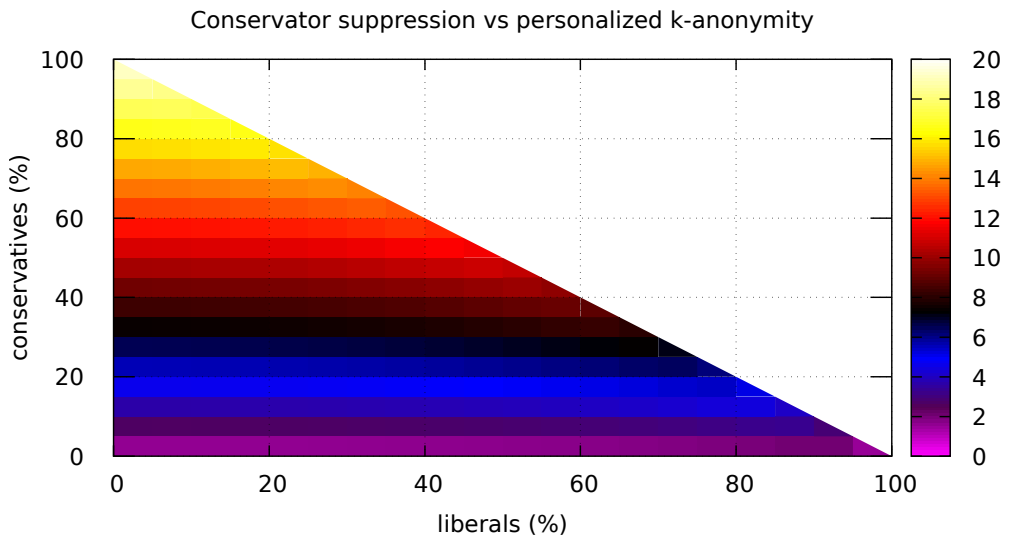


Figure 16: Removing conservators with $k_l = 3$, $k_m = 5$ and $k_c = 7$, referring to figure 5.10a

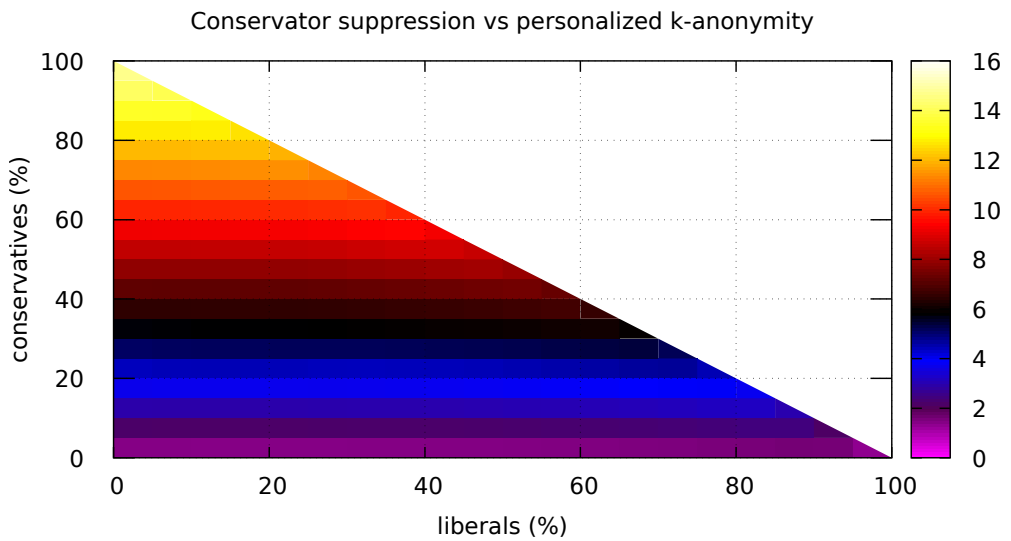


Figure 17: Removing conservators with $k_l = 5$, $k_m = 7$ and $k_c = 10$, referring to figure 5.10b

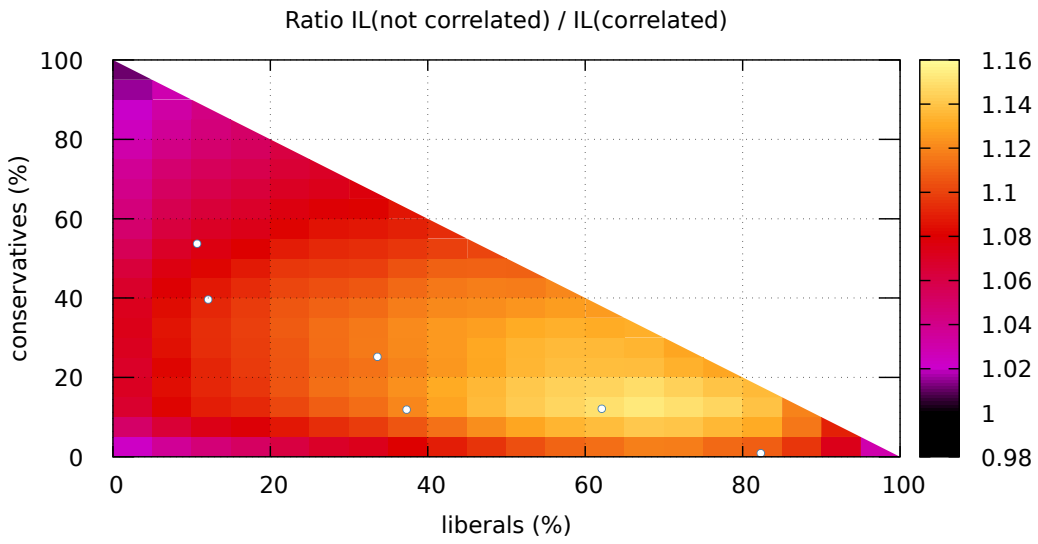


Figure 18: Greedy $k-$Member with low constraints and correlation, referring to figure 5.12a

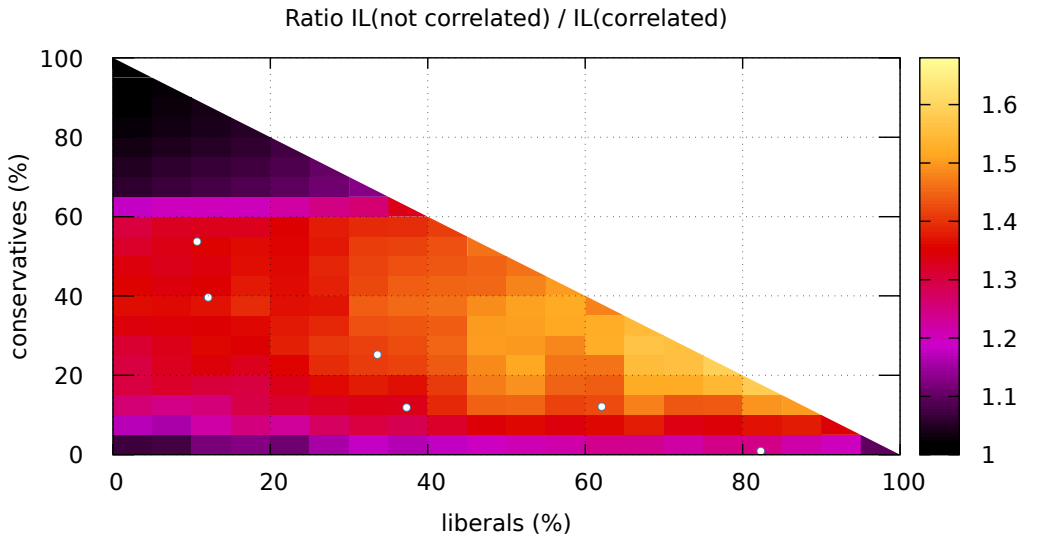


Figure 19: Mondrian with low constraints and correlation, referring to figure 5.12b

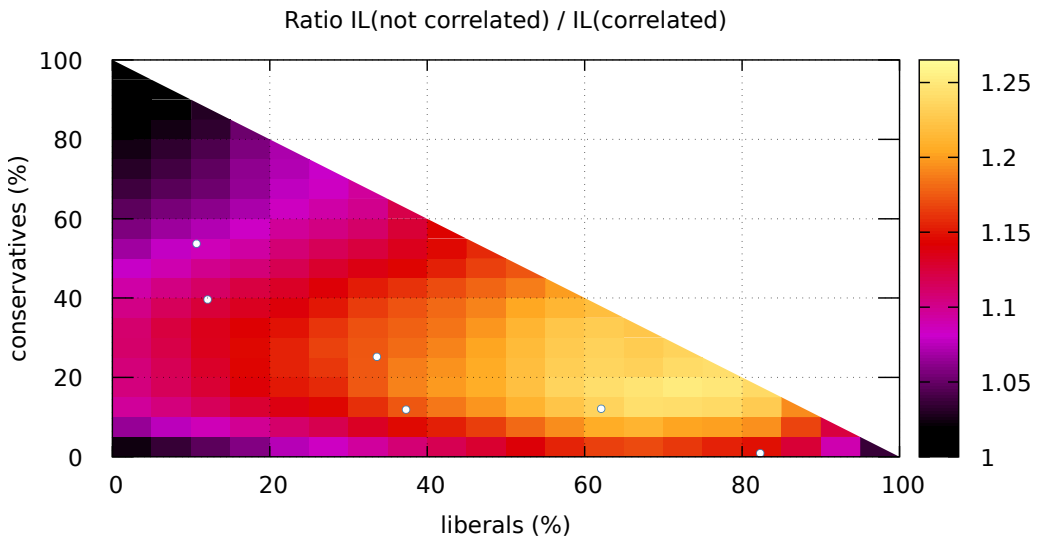


Figure 20: MDAV with low constraints and correlation, referring to figure 5.12c

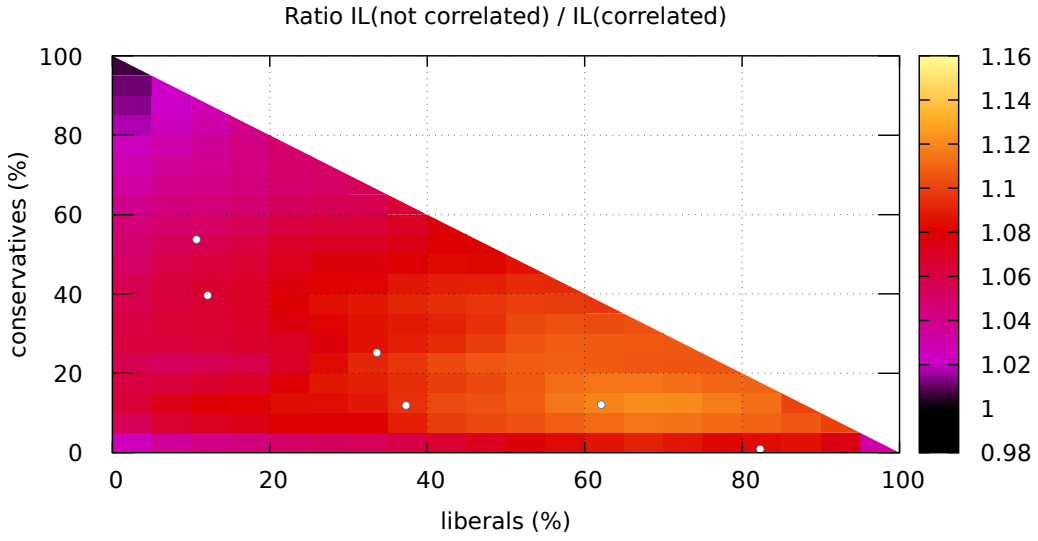


Figure 21: Greedy $k-$Member with high constraints and correlation, referring to figure 5.13a

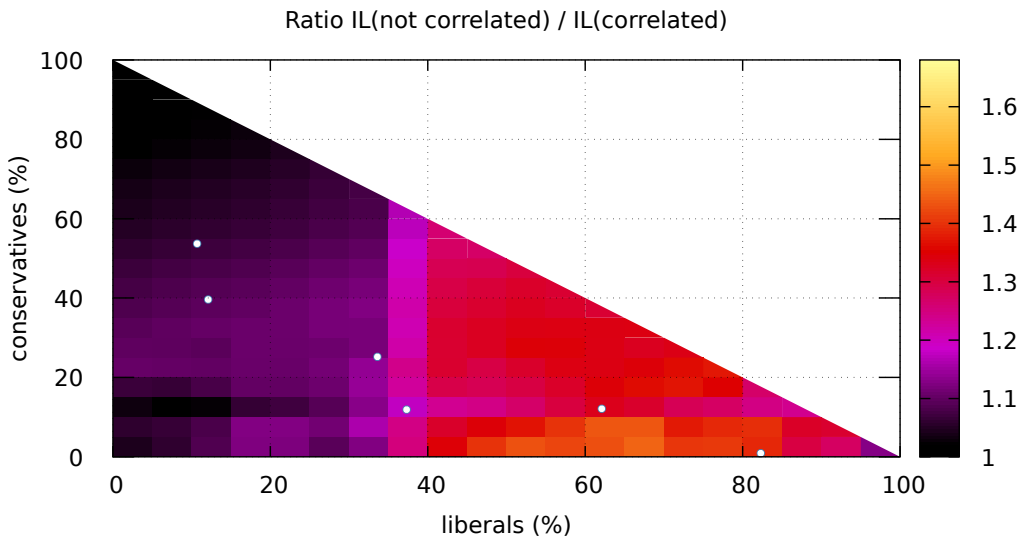


Figure 22: Mondrian with high constraints and correlation, referring to figure 5.13b

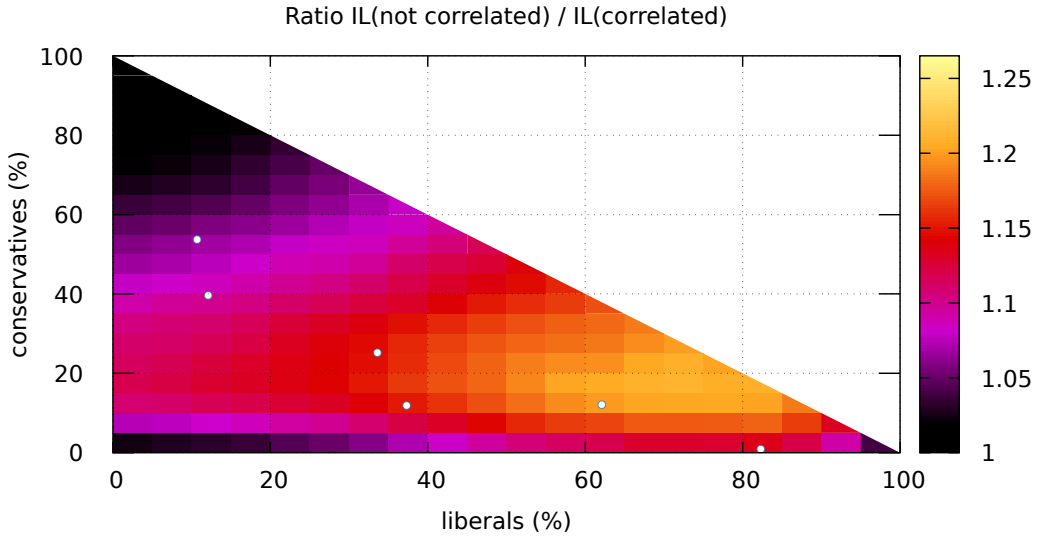


Figure 23: MDAV with high constraints and correlation, referring to figure 5.13c

Bibliography

[1] Serge Abiteboul, Benjamin André, and Daniel Kaplan. "Managing Your Digital Life". In: *Commun. ACM* 58.5 (Apr. 2015), pp. 32–35. ISSN: 0001-0782. DOI: 10.1145/2670528. URL: http://doi.acm.org/10.1145/2670528.

[2] A. Acquisti and J. Grossklags. "Privacy and rationality in individual decision making". In: *IEEE Security Privacy* 3.1 (2005), pp. 26–33. DOI: 10.1109/MSP.2005.22.

[3] T. Allard, B. Nguyen, and P. Pucheral. "Towards a Safe Realization of Privacy-Preserving Data Publishing Mechanisms". In: *2011 IEEE 12th International Conference on Mobile Data Management*. Vol. 2. 2011, pp. 31–34. DOI: 10.1109/MDM.2011.43.

[4] Nicolas Anciaux et al. "Trusted Cells: A Sea Change for Personal Data Services". In: *CIDR 2013, Sixth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 6-9, 2013, Online Proceedings*. 2013.

[5] Miguel E. Andrés et al. "Geo-indistinguishability: Differential Privacy for Location-based Systems". In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13. Berlin, Germany: ACM, 2013, pp. 901–914. ISBN: 978-1-4503-2477-9. DOI: 10.1145/2508859.2516735. URL: http://doi.acm.org/10.1145/2508859.2516735.

[6] S. Bajaj and R. Sion. "TrustedDB: A Trusted Hardware-Based Database with Privacy and Data Confidentiality". In: *IEEE Transactions on Knowledge and Data Engineering* 26.3 (2014), pp. 752–765. ISSN: 1041-4347. DOI: 10.1109/TKDE.2013.38.

[7] Bhuvan Bamba et al. "Supporting Anonymous Location Queries in Mobile Environments with Privacygrid". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 237–246. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367531.

[8] Roberto J. Bayardo and Rakesh Agrawal. "Data Privacy Through Optimal k-Anonymization". In: *Proceedings of the 21st International Conference on Data Engineering*. ICDE '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 217–228. ISBN: 0-7695-2285-8. DOI: 10.1109/ICDE.2005.42. URL: http://dx.doi.org/10.1109/ICDE.2005.42.

[9] N Beldiceanu and E Contejean. "Introducing global constraints in CHIP". In: *Mathematical and Computer Modelling* 20.12 (1994), pp. 97–123. ISSN: 0895-7177. DOI: https://doi.org/10.1016/0895-7177(94)90127-9. URL: http://www.sciencedirect.com/science/article/pii/0895717794901279.

[10] Maxime BERGEAT. *La gestion de la confidentialité pour les données individuelles*. Tech. rep. M2016/07. Institut national de la statistique et des études économiques (INSEE), 2016.

[11] Maxime BERGEAT et al. "A French Anonymization Experiment with Health Data". In: *PSD 2014 : Privacy in Statistical Databases*. Eivissa, Spain, Sept. 2014. URL: https://hal.archives-ouvertes.fr/hal-01214624.

[12] Christian Bessiere et al. "The AllDifferent Constraint with Precedences". In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Ed. by Tobias Achterberg and J. Christopher Beck. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 36–52. ISBN: 978-3-642-21311-3.

[13] Evert Van den Broeck, Karolien Poels, and Michel Walrave. "Older and Wiser? Facebook Use, Privacy Concern, and Privacy Protection in the Life Stages of Emerging, Young, and Middle Adulthood". In: *Social Media + Society* 1.2 (2015), p. 2056305115616149. DOI: 10.1177/2056305115616149.

[14] Ji-Won Byun et al. "Efficient k-Anonymization Using Clustering Techniques". In: *Advances in Databases: Concepts, Systems and Applications*. Ed. by Ramamohanarao Kotagiri et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 188–200. ISBN: 978-3-540-71703-4. DOI: 10.1007/978-3-540-71703-4_18.

[15] Pamara F. Chang et al. "Age Differences in Online Social Networking: Extending Socioemotional Selectivity Theory to Social Network Sites". In: *Journal of Broadcasting & Electronic Media* 59.2 (2015), pp. 221–239. DOI: 10.1080/08838151.2015.1029126.

[16] A. Cuzzocrea, C. Mastroianni, and G. M. Grasso. "Private databases on the cloud: Models, issues and research perspectives". In: *2016 IEEE International Conference on Big Data (Big Data)*. 2016, pp. 3656–3661. DOI: 10.1109/BigData.2016.7841032.

[17] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain. "Constrained clustering by constraint programming". In: *Artificial Intelligence* 244 (2017). Combining Constraint Solving with Mining and Learning, pp. 70 –94. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2015.05.006. URL: http://www.sciencedirect.com/science/article/pii/S0004370215000806.

[18] Josep Domingo-ferrer, Josep M. Mateo-sanz, and Vicenç Torra. "Comparing SDC Methods for Microdata on the Basis of Information Loss and Disclosure". In: *Proceedings of ETK-NTTS 2001, Luxemburg: Eurostat*. Eurostat, 2001, pp. 807–826.

[19] Josep Domingo-Ferrer and Jordi Soria-Comas. "From t-closeness to differential privacy and vice versa in data anonymization". In: *Knowledge-Based Systems* 74 (2015), pp. 151 –158. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2014.11.011. URL: http://www.sciencedirect.com/science/article/pii/S0950705114004031.

[20] Josep Domingo-Ferrer and Vicenç Torra. "A quantitative comparison of disclosure control methods for microdata". In: *Confidentiality, disclosure, and data access : Theory and practical applications for statistical agencies*. Ed. by P. Doyle et al. Elsevier, 2001, pp. 111–133.

[21] Josep Domingo-Ferrer and Vicenç Torra. "Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation". In: *Data ng and Knowledge Discovery* 11.2 (2005), pp. 195–212. ISSN: 1573-756X. DOI: 10.1007/s10618-005-0007-5.

[22] Cynthia Dwork. "Differential Privacy". In: *Proceeding of the 39th International Colloquium on Automata, Languages and Programming*. Vol. 4052. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, pp. 1–12.

[23] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: ACM, 2014, pp. 1054–1067. ISBN: 978-1-4503-2957-6. DOI: 10.1145/2660267.2660348. URL: http://doi.acm.org/10.1145/2660267.2660348.

[24] Martin Ester et al. "A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231. URL: http://dl.acm.org/citation.cfm?id=3001460.3001507.

[25] European Union. *ARTICLE 29 DATA PROTECTION WORKING PARTY: Opinion 05/2014 on Anonymisation Techniques*. 2014.

[26] Philippe Flajolet et al. "Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm". In: *Proceedings of the 2007 International conference on Analysis of Algorithms (AOFA'07)*. 2007.

[27] Paul Francis, Sebastian Probst Eide, and Reinhard Munz. "Diffix: High-Utility Database Anonymization". In: *Privacy Technologies and Policy*. Ed. by Erich Schweighofer et al. Cham: Springer International Publishing, 2017, pp. 141–158. ISBN: 978-3-319-67280-9.

[28] B. C. M. Fung, K. Wang, and P. S. Yu. "Top-down specialization for information and privacy preservation". In: *21st International Conference on Data Engineering (ICDE'05)*. 2005, pp. 205–216. DOI: 10.1109/ICDE.2005.143.

[29] Benjamin C. M. Fung et al. "Privacy-preserving Data Publishing: A Survey of Recent Developments". In: *ACM Comput. Surv.* 42.4 (June 2010), 14:1–14:53. ISSN: 0360-0300. DOI: 10.1145/1749603.1749605. URL: http://doi.acm.org/10.1145/1749603.1749605.

[30] Y. Gahi, M. Guennoun, and K. El-Khatib. "A Secure Database System using Homomorphic Encryption Schemes". In: *ArXiv e-prints* (Dec. 2015). arXiv: 1512.03498 [cs.CR].

[31] S. Gambs, M. Killijian, and M. N. d. P. Cortez. "De-anonymization Attack on Geolocated Data". In: *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. 2013, pp. 789–797. DOI: 10.1109/TrustCom.2013.96.

[32] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. "Show Me How You Move and I Will Tell You Who You Are". In: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. SPRINGL '10. San Jose, California: ACM, 2010, pp. 34–41. ISBN: 978-1-4503-0435-1. DOI: 10.1145/1868470.1868479. URL: http://doi.acm.org/10.1145/1868470.1868479.

[33] Tingjian Ge and Stan Zdonik. "Answering Aggregation Queries in a Secure System Model". In: *Proceedings of the 33rd International Conference on Very Large Data Bases*. VLDB '07. Vienna, Austria: VLDB Endowment, 2007, pp. 519–530. ISBN: 978-1-59593-649-3. URL: http://dl.acm.org/citation.cfm?id=1325851.1325912.

[34] Gecode Team. *Gecode: Generic Constraint Development Environment*. Available from http://www.gecode.org. 2006.

[35] B. Gedik and Ling Liu. "Location Privacy in Mobile Systems: A Personalized Anonymization Model". In: *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. 2005, pp. 620–629. DOI: 10.1109/ICDCS.2005.48.

[36] Craig Gentry. "Fully Homomorphic Encryption Using Ideal Lattices". In: *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: ACM, 2009, pp. 169–178. ISBN: 978-1-60558-506-2. DOI: 10.1145/1536414.1536440. URL: http://doi.acm.org/10.1145/1536414.1536440.

[37] Sean Gilpin and Ian Davidson. "Incorporating SAT Solvers into Hierarchical Clustering Algorithms: An Efficient and Flexible Approach". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. San Diego, California, USA: ACM, 2011, pp. 1136–1144. ISBN: 978-1-4503-0813-7.

DOI: 10.1145/2020408.2020585. URL: http://doi.acm.org/10.1145/2020408.2020585.

[38] A. Gionis, A. Mazza, and T. Tassa. "k-Anonymization Revisited". In: *2008 IEEE 24th International Conference on Data Engineering*. 2008, pp. 744–753. DOI: 10.1109/ICDE.2008.4497483.

[39] Teofilo F. Gonzalez. "Clustering to minimize the maximum intercluster distance". In: *Theoretical Computer Science* 38 (1985), pp. 293–306. ISSN: 0304-3975. DOI: 10.1016/0304-3975(85)90224-5. URL: http://www.sciencedirect.com/science/article/pii/0304397585902245.

[40] Teofilo F. Gonzalez. "Clustering to minimize the maximum intercluster distance". In: *Theoretical Computer Science* 38 (1985), pp. 293–306. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(85)90224-5. URL: http://www.sciencedirect.com/science/article/pii/0304397585902245.

[41] M. Hardt and G. N. Rothblum. "A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis". In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. 2010, pp. 61–70. DOI: 10.1109/FOCS.2010.85.

[42] Anco Hundepool. "The ARGUS Software in the CASC-Project". In: *Privacy in Statistical Databases*. Ed. by Josep Domingo-Ferrer and Vicenç Torra. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 323–335. ISBN: 978-3-540-25955-8. DOI: 10.1007/978-3-540-25955-8_26.

[43] Vijay S. Iyengar. "Transforming Data to Satisfy Privacy Constraints". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. Edmonton, Alberta, Canada: ACM, 2002, pp. 279–288. ISBN: 1-58113-567-X. DOI: 10.1145/775047.775089. URL: http://doi.acm.org/10.1145/775047.775089.

[44] L.C.R.J. Willenborg J.M. Gouweleeuw P. Kooiman and P.-P. de Wolf. "Post Randomisation for Statistical Disclosure Control: Theory and Implementation". In: *Journal of Official Statistics* 14.4 (1998), pp. 463–478. ISSN: 0282-423X.

[45] Z. Jorgensen, T. Yu, and G. Cormode. "Conservative or liberal? Personalized differential privacy". In: *2015 IEEE 31st International Conference on Data Engineering*. 2015, pp. 1023–1034. DOI: 10.1109/ICDE.2015.7113353.

[46] Pekka Jousilahti et al. "Sex, age, cardiovascular risk factors, and coronary heart disease: a prospective follow-up study of 14 786 middle-aged men and women in Finland". In: *Circulation* 99.9 (1999), pp. 1165–1172.

[47] Athanasia Katsouraki. "Sharing and Usage Control of Personal Information. (Partage et Contrôle d'Usage de Données Personnelles)". PhD thesis. University of Paris-Saclay, France, 2016. URL: https://tel.archives-ouvertes.fr/tel-01425638.

[48] Murat Kezer et al. "Age differences in privacy attitudes, literacy and privacy management on Facebook". In: vol. 10. 1. 2016. DOI: 10.5817/CP2016-1-2.

[49] Edward C. Kokkelenberg, Michael Dillon, and Sean M. Christy. "The effects of class size on student grades at a public university". In: *Economics of Education Review* 27.2 (2008), pp. 221 –233. ISSN: 0272-7757. DOI: https://doi.org/10.1016/j.econedurev.2006.09.011. URL: http://www.sciencedirect.com/science/article/pii/S0272775707000271.

[50] Mikael Z. Lagerkvist and Christian Schulte. "Advisors for Incremental Propagation". In: *Principles and Practice of Constraint Programming – CP 2007*. Ed. by Christian Bessière. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 409–422. ISBN: 978-3-540-74970-7.

[51] Saliha Lallali et al. "A Secure Search Engine for the Personal Cloud". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD '15. Melbourne, Victoria, Australia: ACM, 2015, pp. 1445–1450. ISBN: 978-1-4503-2758-9. DOI: 10.1145/2723372.2735376. URL: http://doi.acm.org/10.1145/2723372.2735376.

[52] Nadjib Lazaar et al. "A Global Constraint for Closed Frequent Pattern Mining". In: *Principles and Practice of Constraint Programming*. Ed. by Michel Rueher. Cham: Springer International Publishing, 2016, pp. 333–349. ISBN: 978-3-319-44953-1.

[53] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. "Incognito: Efficient Full-domain K-anonymity". In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. SIGMOD '05. Baltimore, Maryland: ACM, 2005, pp. 49–60. ISBN: 1-59593-060-4. DOI: 10.1145/1066157.1066164. URL: http://doi.acm.org/10.1145/1066157.1066164.

[54] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. "Mondrian Multidimensional K-Anonymity". In: *Proceedings of the 22Nd International Conference on Data Engineering*. ICDE '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 25–. ISBN: 0-7695-2570-9. DOI: 10.1109/ICDE.2006.101.

[55] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. "Workload-aware Anonymization". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. Philadelphia, PA, USA: ACM, 2006, pp. 277–286. ISBN: 1-59593-339-5. DOI: 10.1145/1150402.1150435. URL: http://doi.acm.org/10.1145/1150402.1150435.

[56] Haoran Li et al. "Partitioning-Based Mechanisms Under Personalized Differential Privacy". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jinho Kim et al. Cham: Springer International Publishing, 2017, pp. 615–627. ISBN: 978-3-319-57454-7.

[57] Jiuyong Li et al. "Achieving k-Anonymity by Clustering in Attribute Hierarchical Structures". In: *Data Warehousing and Knowledge Discovery*. Ed. by A. Min Tjoa and Juan Trujillo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 405–416. ISBN: 978-3-540-37737-5.

[58] N. Li, T. Li, and S. Venkatasubramanian. "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity". In: *2007 IEEE 23rd International Conference on Data Engineering*. 2007, pp. 106–115. DOI: 10.1109/ICDE.2007.367856.

[59] M. Lichman. *UCI Machine Learning Repository*. 2013. URL: http://archive.ics.uci.edu/ml.

[60] X. Liu, Q. Xie, and L. Wang. "A Personalized Extended (a, k)-Anonymity Model". In: *2015 Third International Conference on Advanced Cloud and Big Data*. 2015, pp. 234–240. DOI: 10.1109/CBD.2015.45.

[61] Ashwin Machanavajjhala et al. "l-Diversity: Privacy Beyond k-Anonymity". In: *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*. 2006, p. 24. DOI: 10.1109/ICDE.2006.1.

[62] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

[63] F. McSherry and K. Talwar. "Mechanism Design via Differential Privacy". In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. 2007, pp. 94–103. DOI: 10.1109/FOCS.2007.66.

[64] Frank D. McSherry. "Privacy Integrated Queries: An Extensible Platform for Privacy-preserving Data Analysis". In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. SIGMOD '09. Providence, Rhode Island, USA: ACM, 2009, pp. 19–30. ISBN: 978-1-60558-551-2. DOI: 10.1145/1559845.1559850. URL: http://doi.acm.org/10.1145/1559845.1559850.

[65] Axel Michel and Benjamin Nguyen. "Exécution de requêtes distribuées sous contraintes d'anonymat. Sur l'architecture des Trusted Cells". In: *Gestion des risques naturels, technologiques et sanitaires*. Envirorisk. Bourges, France: Cépaduès, 2016. ISBN: 978-2-36493-549-5. URL: https://www.cepadues.com/livres/Gestion-des-risques-naturels,-technologiques-et-sanitaires-(congr%C3%A8s-Envirorisk-2016)-9782364935495.html.

[66] Axel Michel, Benjamin Nguyen, and Philippe Pucheral. "Managing Distributed Queries under Personalized Anonymity Constraints". In: *Proceedings of the 6th International Conference on Data Science, Technology and Applications - Volume 1: DATA,* INSTICC. SciTePress, 2017, pp. 107–117. ISBN: 978-989-758-255-4. DOI: 10.5220/0006477001070117.

[67] Axel Michel, Benjamin Nguyen, and Philippe Pucheral. "The Case for Personalized Anonymization of Database Query Results". In: *Data Management Technologies and Applications*. Ed. by Joaquim Filipe, Jorge Bernardino, and Christoph Quix. Cham: Springer International Publishing, 2018, pp. 261–285. ISBN: 978-3-319-94809-6.

[68] Noman Mohammed et al. "Differentially Private Data Release for Data Mining". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. San Diego, California, USA: ACM, 2011, pp. 493–501. ISBN: 978-1-4503-0813-7. DOI: 10.1145/2020408.2020487. URL: http://doi.acm.org/10.1145/2020408.2020487.

[69] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. "The New Casper: Query Processing for Location Services Without Compromising Privacy". In: *Proceedings of the 32Nd International Conference on Very Large Data Bases*. VLDB '06. Seoul, Korea: VLDB Endowment, 2006, pp. 763–774. URL: http://dl.acm.org/citation.cfm?id=1182635.1164193.

[70] Barun Kumar Nayak. "Understanding the relevance of sample size calculation". In: *Indian J Ophthalmol* 58.6 (2010), pp. 469–470. ISSN: 0301-4738. DOI: 10.4103/0301-4738.71673. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2993974/.

[71] Sang Ni, Mengbo Xie, and Quan Qian. "Clustering Based K-anonymity Algorithm for Privacy Preservation". In: *International Journal of Network Security* 19.6 (2017), pp. 1062–1071. ISSN: 1816-353X. DOI: 10.6633/IJNS.201711.19(6).23.

[72] Anna Oganian and Josep Domingo-ferrer. "On the Complexity of Optimal Microaggregation for Statistical Disclosure Control". In: *Statistical Journal of the United Nations Economic Comission for Europe* 18 (2001), pp. 345–354.

[73] Pascal Paillier. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". In: *Advances in Cryptology — EUROCRYPT '99*. Ed. by Jacques Stern. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238. ISBN: 978-3-540-48910-8.

[74] Raluca Ada Popa et al. "CryptDB: Protecting Confidentiality with Encrypted Query Processing". In: *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. SOSP '11. Cascais, Portugal: ACM, 2011, pp. 85–100. ISBN: 978-1-4503-0977-6. DOI: 10.1145/2043556.2043566. URL: http://doi.acm.org/10.1145/2043556.2043566.

[75] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance).* URL: http://data.europa.eu/eli/reg/2016/679/oj.

[76] Aaron Roth and Tim Roughgarden. "Interactive Privacy via the Median Mechanism". In: *Proceedings of the Forty-second ACM Symposium on Theory of Computing*. STOC '10. Cambridge, Massachusetts, USA: ACM, 2010, pp. 765–774. ISBN: 978-1-4503-0050-6. DOI: 10.1145/1806689.1806794. URL: http://doi.acm.org/10.1145/1806689.1806794.

[77] Pierangela Samarati and Latanya Sweeney. "Generalizing Data to Provide Anonymity when Disclosing Information (Abstract)". In: *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. PODS '98. Seattle, Washington, USA: ACM, 1998, pp. 188–. ISBN: 0-89791-996-3. DOI: 10.1145/275487.275508. URL: http://doi.acm.org/10.1145/275487.275508.

[78] Y. Shen, Y. Liu, and Y. Zhang. "Personalized-Granular k-Anonymity". In: *2009 International Conference on Information Engineering and Computer Science*. 2009, pp. 1–4. DOI: 10.1109/ICIECS.2009.5365939.

[79] H. Jeff Smith, Sandra J. Milberg, and Sandra J. Burke. "Information Privacy: Measuring Individuals' Concerns about Organizational Practices". In: *MIS Quarterly* 20.2 (1996), pp. 167–196. ISSN: 02767783. DOI: 10.2307/249477. URL: http://www.jstor.org/stable/249477.

[80] Louis Philippe Sondeck, Maryline Laurent, and Vincent Frey. "Discrimination rate: an attribute-centric metric to measure privacy". In: *Annals of Telecommunications* 72.11 (2017), pp. 755–766. ISSN: 1958-9395. DOI: 10.1007/s12243-017-0581-8. URL: https://doi.org/10.1007/s12243-017-0581-8.

[81] Louis Philippe Sondeck, Maryline Laurent, and Vincent Frey. "The Semantic Discrimination Rate Metric for Privacy Measurements which Questions the Benefit of t-closeness over l-diversity". In: *Proceedings of*

the 14th International Joint Conference on e-Business and Telecommunications - Volume 6: SECRYPT, (ICETE 2017). INSTICC. SciTePress, 2017, pp. 285–294. ISBN: 978-989-758-259-2. DOI: 10.5220/0006418002850294.

[82] Latanya Sweeney. "Achieving K-anonymity Privacy Protection Using Generalization and Suppression". In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10.5 (Oct. 2002), pp. 571–588. ISSN: 0218-4885. DOI: 10.1142/S021848850200165X. URL: http://dx.doi.org/10.1142/S021848850200165X.

[83] Latanya Sweeney. "k-Anonymity: A Model for Protecting Privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.5 (2002), pp. 557–570. DOI: 10.1142/S0218488502001648.

[84] Matthias Templ, Alexander Kowarik, and Bernhard Meindl. "Statistical Disclosure Control for Micro-Data Using the R Package sdcMicro". In: *Journal of Statistical Software* 67.4 (2015), pp. 1–36. DOI: 10.18637/jss.v067.i04.

[85] Quoc-Cuong To, Benjamin Nguyen, and Philippe Pucheral. "Private and Scalable Execution of SQL Aggregates on a Secure Decentralized Architecture". In: *ACM Trans. Database Syst.* 41.3 (Aug. 2016), 16:1–16:43. ISSN: 0362-5915. DOI: 10.1145/2894750. URL: http://doi.acm.org/10.1145/2894750.

[86] Quoc-Cuong To, Benjamin Nguyen, and Philippe Pucheral. "SQL/AA: Executing SQL on an Asymmetric Architecture". In: *PVLDB* 7.13 (2014), pp. 1625–1628. URL: http://www.vldb.org/pvldb/vol7/p1625-to.pdf.

[87] H. K. Tsoi and L. Chen. "From Privacy Concern to Uses of Social Network Sites: A Cultural Comparison via User Survey". In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. 2011, pp. 457–464. DOI: 10.1109/PASSAT/SocialCom.2011.71.

[88] A. K. H. Tung, J. Hou, and Jiawei Han. "Spatial clustering in the presence of obstacles". In: *Proceedings 17th International Conference on Data Engineering*. 2001, pp. 359–367. DOI: 10.1109/ICDE.2001.914848.

[89] Howard K. Wachtel. "Student Evaluation of College Teaching Effectiveness: a brief review". In: *Assessment & Evaluation in Higher Education* 23.2 (1998), pp. 191–212. DOI: 10.1080/0260293980230207. URL: https://doi.org/10.1080/0260293980230207.

[90] Ke Wang, P. S. Yu, and S. Chakraborty. "Bottom-up generalization: a data mining solution to privacy protection". In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*. 2004, pp. 249–256. DOI: 10.1109/ICDM.2004.10110.

[91] Daniela M. Witten and Robert Tibshirani. "A Framework for Feature Selection in Clustering". In: *Journal of the American Statistical Association* 105.490 (2010). PMID: 20811510, pp. 713–726. DOI: 10.1198/jasa.2010.tm09415. URL: https://doi.org/10.1198/jasa.2010.tm09415.

[92] Raymond Chi-Wing Wong et al. "(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*. KDD '06. Philadelphia, PA, USA: ACM, 2006, pp. 754–759. ISBN: 1-59593-339-5. DOI: 10.1145/1150402.1150499. URL: http://doi.acm.org/10.1145/1150402.1150499.

[93] Raymond Chi-Wing Wong et al. "Minimality Attack in Privacy Preserving Data Publishing". In: *Proceedings of the 33rd International Conference on Very Large Data Bases*. VLDB '07. Vienna, Austria: VLDB Endowment, 2007, pp. 543–554. ISBN: 978-1-59593-649-3. URL: http://dl.acm.org/citation.cfm?id=1325851.1325914.

[94] Zhibiao Wu and Martha Palmer. "Verbs Semantics and Lexical Selection". In: *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*. ACL '94. Las Cruces, New Mexico: Association for Computational Linguistics, 1994, pp. 133–138. DOI: 10.3115/981732.981751. URL: https://doi.org/10.3115/981732.981751.

[95] Xiaokui Xiao and Yufei Tao. "Personalized Privacy Preservation". In: *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*. SIGMOD '06. Chicago, IL, USA: ACM, 2006, pp. 229–240. ISBN: 1-59593-434-0. DOI: 10.1145/1142473.1142500.

[96] Xiaokui Xiao et al. "iReduct: Differential Privacy with Reduced Relative Errors". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. SIGMOD '11. Athens, Greece: ACM, 2011, pp. 229–240. ISBN: 978-1-4503-0661-4. DOI: 10.1145/1989323.1989348. URL: http://doi.acm.org/10.1145/1989323.1989348.

[97] J. Xu et al. "Differentially Private Histogram Publication". In: *2012 IEEE 28th International Conference on Data Engineering*. 2012, pp. 32–43. DOI: 10.1109/ICDE.2012.48.

[98] J. Yu et al. "TopDown-KACA: An efficient local-recoding algorithm for k-anonymity". In: *2009 IEEE International Conference on Granular Computing*. 2009, pp. 727–732. DOI: 10.1109/GRC.2009.5255024.

[99] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. "BIRCH: An Efficient Data Clustering Method for Very Large Databases". In: *SIGMOD Rec.* 25.2 (June 1996), pp. 103–114. ISSN: 0163-5808. DOI: 10.1145/235968.233324. URL: http://doi.acm.org/10.1145/235968.233324.

[100] T. Zhu et al. "Differentially Private Data Publishing and Analysis: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* 29.8 (2017), pp. 1619–1638. ISSN: 1041-4347. DOI: 10.1109/TKDE.2017.2697856.

[Axel MICHEL]

Personnalisation de protection de la vie privée sur des modèles d'anonymisation basés sur des généralisations

Résumé : Les bénéfices engendrés par les études statistiques sur les données personnelles des individus sont nombreux, que ce soit dans le médical, l'énergie ou la gestion du trafic urbain pour n'en citer que quelques-uns. Les initiatives publiques de smart-disclosure et d'ouverture des données rendent ces études statistiques indispensables pour les institutions et industries tout autour du globe. Cependant, ces calculs peuvent exposer les données personnelles des individus, portant ainsi atteinte à leur vie privée. Les individus sont alors de plus en plus réticent à participer à des études statistiques malgré les protections garanties par les instituts. Pour retrouver la confiance des individus, il devient nécessaire de proposer des solutions de user empowerment, c'est-à-dire permettre à chaque utilisateur de contrôler les paramètres de protection des données personnelles les concernant qui sont utilisées pour des calculs.

Cette thèse développe donc un nouveau concept d'anonymisation personnalisé, basé sur la généralisation de données et sur le user empowerment.

En premier lieu, ce manuscrit propose une nouvelle approche mettant en avant la personnalisation des protections de la vie privée par les individus, lors de calculs d'agrégation dans une base de données. De cette façon les individus peuvent fournir des données de précision variable, en fonction de leur perception du risque. De plus, nous utilisons une architecture décentralisée basée sur du matériel sécurisé assurant ainsi les garanties de respect de la vie privée tout au long des opérations d'agrégation.

En deuxième lieu, ce manuscrit étudie la personnalisations des garanties d'anonymat lors de la publication de jeux de données anonymisés. Nous proposons l'adaptation d'heuristiques existantes ainsi qu'une nouvelle approche basée sur la programmation par contraintes. Des expérimentations ont été menées pour étudier l'impact d'une telle personnalisation sur la qualité des données. Les contraintes d'anonymat ont été construites et simulées de façon réaliste en se basant sur des résultats d'études sociologiques.

Mots clés : Protection de la vie privée et des données, Big Data, Matériel sécurisé, Programmation par contraintes

Personalizing Privacy Constraints in Generalization-based Anonymization Models

Summary : The benefit of performing Big data computations over individual's microdata is manifold, in the medical, energy or transportation fields to cite only a few, and this interest is growing with the emergence of smart-disclosure initiatives around the world.However, these computations often expose microdata to privacy leakages, explaining the reluctance of individuals to participate in studies despite the privacy guarantees promised by statistical institutes. To regain indivuals'trust, it becomes essential to propose user empowerment solutions, that is to say allowing individuals to control the privacy parameter used to mke computations over their microdata.

This work proposes a novel concept of personalized anonymisation based on data generalization and user empowerment.

Firstly, this manuscript proposes a novel approach to push personalized privacy guarantees in the processing of database queries so that individuals can disclose different amounts of information (i.e. data at different levels of accuracy) depending on their own perception of the risk. Moreover, we propose a decentralized computing infrastructure based on secure hardware enforcing these personalized privacy guarantees all along the query execution process.

Secondly, this manuscript studies the personalization of anonymity guarantees when publishing data. We propose the adapation of existing heuristics and a new approach based on constraint programming. Experiments have been done to show the impact of such personalization on the data quality. Individuals'privacy constraints have been built and realistically using social statistic studies.

Keywords: Data privacy and security, Big Data, Secure Hardware, Constraint Programming

LABORATOIRE D'INFORMATIQUE FONDAMENTALE D'ORLEANS

INSA Centre Val de Loire
88 Boulevard Lahitolle, 18000 Bourges

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE