Exposition Minimum de Données pour des Applications à Base de Classifieurs

Nicolas Anciaux[1,2], Benjamin Nguyen[1,2], Michalis Vazirgiannis[3,4]

1. *Laboratoire PRiSM, CNRS UMR 8144, 78035 Versailles*
2. *INRIA Rocquencourt, 78153 Le Chesnay*
3. *Athens University of Economics and Business, GR10434 Athènes, Grèce*
4. *Laboratoire d'Informatique de l'X, CNRS UMR 7161, 92218 Palaiseau*

RESUME. — Les formulaires d'application sont souvent utilisés pour collecter des données personnelles sur les postulants, et par la suite pour ajuster ces services à leur situation spécifique. L'ensemble de ces données doit être réduit à son strict minimum en vue du traitement ultérieur. Actuellement, il n'existe aucune technique permettant cette minimisation pour des applications complexes d'aide à la décision basées sur des classifieurs. Dans cet article, nous étendons nos résultats précédents, en présentant le problème généralisé d'Exposition Minimale des données. Nous montrons que ce problème est NP-difficile, et discutons de sa résolution. Les expériences montrent que la réduction des données transmises en utilisant notre technique est importante.

ABSTRACT. Application forms are often used to collect personal information, and subsequently to tailor these services to the applicant's specific situation. The data collected must be reduced to a minimum size in order to serve the objective process. However, no technique exists today to minimize information used by complex decision processes based on classifiers. In this article, we extend our previous results by presenting the generalized Minimum Exposure problem. We show that this problem is NP-hard and discuss its resolution. Experiments show a big reduction in data transmitted by using our technique..

MOTS-CLES : Principe de protection de la vie privée ; Collection limitée ; remplissage automatique de formulaires.

KEYWORDS: Privacy principle; Limited collection; Automated form filling.

DOI:10.3199/TSI.31.1-n © 2012 Lavoisier AR_DOI

1. Introduction

Nous vivons actuellement une numérisation massive des données personnelles. Nous recevons sous format numérique une quantité croissante de documents

importants (financiers, professionnels, médicaux, d'assurance, administratifs, etc.). Ces documents sont produits par nos employeurs, nos banques, nos assurances, les services de l'Etat, les hôpitaux, les écoles, les FAI, et compagnies de télécoms, etc. Parallèlement, le domaine du *nuage personnel (Personal Cloud)* est florissant, un rapport récent estimant ce marché à près de 12 milliards de dollars[1], avec des offres telles qu'Adminium[2] ou Securibox[3]. Des offres alternatives proposent des moyens de stockage côté utilisateur, augmentant ainsi leur contrôle sur la dissémination de ces informations personnelles, comme par exemple les Serveurs de Données Personnelles, dits *Personal Data Servers* dans Allard *et al.* (2010), the Personal Data Ecosystem[4], Nori[5] ou encore des *Plug Servers* comme la FreedomBox[6].

L'explosion de ce marché s'explique par la tendance suivante : les documents officiels sont accumulés et conservés précieusement par leur possesseur, pour des raisons légales d'obligation de conservation (e.g. 1 an pour des relevés bancaires). Ces documents sont également utilisés comme preuves ou attestations lors de procédures administratives (e.g. impôts) ou de demandes de services (e.g. prêt).

Dans cet article, nous prenons comme exemple une interaction entre un postulant et un fournisseur de service, où ce fournisseur réclame des données personnelles supplémentaires sur le postulant, pour pouvoir lui présenter la meilleure offre. Parmi de telles interactions, on pourra citer le calibrage de services pour prendre en compte la situation particulière du postulant. Par exemple, les caractéristiques d'un prêt personnel (taux, durée, assurance, etc.) peuvent être définies par un processus de prise de décision utilisant des données personnelles telles que le revenu, le fait de posséder un CDI, des titres de propriété, des informations de santé, des garanties, un historique de remboursement de crédits, etc. On peut également citer d'autres exemples tels que les contrats d'assurances, les aides sociales, d'une manière général, toute information personnelle décrivant la situation spécifique du postulant, et qui permettrait de ce calibrer l'offre au mieux.

Ainsi, nous ne remettons pas en cause la nécessité d'évaluer la situation personnelle particulière du postulant, et ce à la fois dans son propre intérêt mais aussi dans celui du fournisseur de service. Toutefois, l'ensemble des informations personnelles requises doit être réduit à son strict minimum pour deux raisons principales. En premier lieu, il faut garantir la protection de la vie privée du postulant. Partout dans le monde, de nombreux textes de lois ont été votés (EP, 95 ; OCDE, 1980) qui suggèrent et introduisent à cet égard le principe de Collecte Minimale des Données (*CMD*) qui affirme que les données personnelles collectées doivent être minimales pour atteindre l'objectif proposé et auquel l'utilisateur a consenti. En second lieu, le coût d'une fuite d'informations doit être réduit. En effet, bien trop souvent, des données personnelles finissent par être dévoilées, à cause de

[1] The Personal Cloud: Transforming Personal Computing, Mobile, And Web Markets, Frank Gillett, a Forrester report, June 2011.
[2] Voir http://www.adminium.fr/
[3] Voir http://securibox.fr/
[4] Voir http://personaldataecosystem.org/
[5] Voir http://www.projectnori.org/
[6] Voir http://freedomboxfoundation.org/

négligences ou d'attaques. En 2011, la *Open Security Foundation*[7] a recueilli plus d'un millier d'incidents affectant plus d'une centaine de millions d'enregistrements. La *Privacy Rights Clearinghouse*[8] a étudié 275 brêches affectant plus de 20 millions d'enregistrements. Chaque incident est un désastre financier pour les entreprises gérant les données. Une étude récente (Ponemon, 2011) estime le coût d'une fuite d'information à 7,2 millions de dollars en moyenne par incident, qui a subit un accroissement important entre 2006 et 2012. Ce coût provient de plusieurs facteurs : des lois passées dans de nombreux pays, dans l'union européenne[9], et dans 46 états américains[10] obligent les entreprises victimes de fuites d'informations à notifier les clients dont les données ont été perdues, à les aider à minimiser les conséquences de cette perte (e.g. annuler leur carte bancaire si son code a été dévoilé), et souvent à leur fournir un dédommagement financier. Des entreprises de sécurité informatique proposent en ligne des calculateurs de coût d'une fuite d'information[11] pour attirer l'attention du public sur ce problème : plus grande est la quantité de données exposées, plus grand est le coût d'une fuite d'informations.

L'objectif de ce papier est de proposer une méthode pour réduire la taille de l'ensemble des informations exposées par les utilisateurs aux fournisseurs de services, en accord avec le principe de *CMD*, tout en ayant aucun impact sur l'évaluation finale du dossier, conséquence du processus de prise de décision.

Ce problème est difficile. Certains travaux déjà existants ont essayé de transposer des principes légaux aux systèmes d'informations, comme par exemple les bases de données Hippocratiques (Agrawal *et al.* 2002), qui maintiennent l'ensemble des attributs nécessaires pour atteindre un objectif donné. Cependant, en général il est impossible de déterminer *a priori* (au moment de la collecte) si une information est utile ou inutile pour la prise de décision. Une telle hypothèse n'est en effet valable que pour des cas très simples (e.g. lors d'une commande en ligne, l'adresse du client est nécessaire pour la livraison du produit). Toutefois dans le cas général d'un système d'aide à la décision cette hypothèse est fausse. Quelle donnée est en effet nécessaire pour justifier la réduction d'un taux proposé à un utilisateur ? La collecte d'information dépend donc à la fois de l'objectif, mais aussi des données elles-mêmes. Prenons l'exemple suivant d'une réduction de taux dépendant de l'âge, du salaire et du patrimoine d'une personne. Montrer un salaire de $*30.000* par an si la personne est plus jeune que 25 ans pourrait être suffisant. Un salaire de $*50.000* par an conviendrait également, quel que soit l'âge du client. Ou encore, l'âge et le salaire pourraient ne pas être important si le client possède un patrimoine supérieur à $*100.000*. Pour un utilisateur défini par le tuple u_1=[*salaire=$35.000*, *âge=21*, *patrimoine=$10.000*] l'ensemble minimum serait [*salaire*, *âge*]. Pour un utilisateur défini par u_2=[*salaire=$40.000*, *age=35*, *patrimoine=$250.000*], ce serait [*patrimoine*]. Ainsi, une banque ne pourrait pas définir un ensemble minimum

[7] Voir http://www.datalossdb.org/reports
[8] Voir http://www.privacyrights.org/
[9] UE Résolution du 6 mai 2009
[10] http://www.ncsl.org/issues-research/telecommunications-information-technology/security-breach-notification-laws.aspx
[11] Voir http://databreachcalculator.com.sapin.arvixe.com/

d'attributs à collecter pour prendre sa décision, puisque déterminer l'ensemble minimum nécessite de consulter l'ensemble des attributs. La définition *a priori* d'un ensemble de données à collecter conduit toujours à surévaluer sa taille.

La procédure habituelle suivie par les fournisseurs de service qui utilisent un processus d'aide à la décision est de demander aux utilisateurs de remplir des formulaires contenant toutes les informations *qui pourraient être nécessaires* dans le cadre du processus. Cette procédure n'est évidement (et par définition) pas compatible avec le principe de *CMD*, puisque les fournisseurs de services collectent des données qui *pourraient n'avoir aucun impact* sur la décision finale: c'est le *paradoxe* de la collecte limitée de données: *les systèmes de gestion de données ont évidemment besoin d'accéder aux données pour savoir si ces données leur seront utiles.*

Notre approche est basée sur une implémentation inverse de la stratégie de *CMD* classique dans laquelle il faudra fournir suffisamment d'informations sur le processus sous-jacent de prise de décision pour permettre à l'utilisateur de déterminer localement l'ensemble minimum de données à fournir afin d'obtenir le service demandé avec un bénéfice maximal.

L'article est organisé de la manière suivante : la Section 2 décrit le scénario général, et présente un exemple utilisé tout au long du papier. Dans la Section 3, nous énonçons le problème d'optimisation que nous nommons *Exposition Minimum*, et étudions sa complexité. Plusieurs algorithmes de résolution sont proposés en Section 4, et validés par une expérimentation présentée en Section 5. La Section 6 traite des travaux connexes, et la Section 7 conclut.

Les contributions spécifiques de cet article par rapport à (Anciaux *et al.*, 2011) sont (i) la généralisation du problème d'Exposition Minimale au cas où l'utilisateur possède plusieurs documents permettant de prouver un même attribut, et (ii) la preuve formelle de complexité du problème. Afin que l'article soit autosuffisant, nous redéfinissons les notations d'Anciaux *et al.* (2011).

2. Scénario d'Exposition Minimum

2.1. Scenario Général

Nous considérons le scénario générique illustré par la Figure 1, qui contient trois acteurs principaux : les Producteurs de Données, les Utilisateurs, et les Fournisseurs de Services. Les Producteurs de Données jouent le rôle de sources de données. Ils incluent par exemple des banques, des employeurs, des hôpitaux ou des administrations. Ils fournissent aux utilisateurs des informations qui peuvent être signées pour prouver leur intégrité et leur authenticité (e.g. bulletins de paie, relevés de comptes, factures, etc.). Les Utilisateurs stockent ces documents reçus dans leur espace de stockage personnel. Nous ne faisons aucune hypothèse sur cet espace, qui peut tout aussi bien être leur propre PC qu'un système de stockage sur le nuage, ou sur un appareil utilisant du matériel sécurisé. Les Fournisseurs de Services incluent des compagnies privées, telles que des banques ou des assurances, mais aussi des administrations publiques, comme les services sociaux. Ils proposent des services

tels que des prêts, des assurances ou des aides sociales, qui nécessitent des données personnelles pour prendre une décision et calibrer l'offre à l'utilisateur postulant.

En pratique, les Fournisseurs de Services émettent des formulaires pour collecter les données pouvant influencer leur décision. Dans certains contextes, d'énormes quantités de données peuvent être requises par les formulaires. Par exemple, des formulaires d'application à des prêts immobiliers recueillent communément des centaines de données personnelles[12]. Les aides sociales comme le GEVA requièrent également de remplir un formulaire avec près de 500 champs.

Nous promouvons dans ce papier une nouvelle approche, où le Fournisseur de Services doit fournir non seulement le formulaire, mais aussi un ensemble de *règles de collection*. Ces règles vont permettre aux utilisateurs de sélectionner, au sein des informations potentiellement requises par le formulaire, un sous-ensemble minimum à remplir. Nous appelons *Exposition Minimum* (*EM*) le processus permettant d'identifier un ensemble minimum *d'assertions* à exposer au Fournisseur de Service (i.e. de champs à remplir par l'utilisateur) pour obtenir le service requis ayant l'intégralité des bénéfices auquel l'utilisateur peut (et souhaite) prétendre.

EM nécessite de confronter l'ensemble des assertions qui peuvent être publiées par l'Utilisateur, aux règles de collecte décrivant l'information requise par le Fournisseur de Service pour lui donner un bénéfice spécifique.

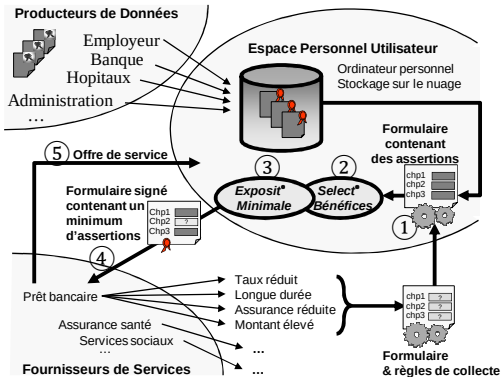

Figure 1. Architecture Générique supportant l'Exposition Minimum

L'exécution de *EM* doit se dérouler côté utilisateur, ou sur un tiers de confiance, afin de respecter le principe de *CMD*. En effet, le système responsable de l'exécution de *EM* doit collecter plus de données qu'un sous-ensemble minimum finalement calculé par *EM*.

[12] Voir le formulaire de la *Nationwide Building Society* (plus grande entreprise de prêt immobilier mondiale) comme un exemple typique: http://www.nationwide.co.uk/nr/rdonlyres/a48ffc87-7e29-4ea6-b24d-2720746c5d9e/0/m1inov06.pdf

Le scénario général illustré par la Figure 1 est le suivant : lorsqu'un utilisateur veut postuler à un service, ① il télécharge le formulaire et les règles de collecte fournis par le Fournisseur de Service, et remplit un maximum de champs dans ce formulaire en utilisant les documents à sa disposition; ② il utilise le *Sélectionneur de Bénéfices* pour calculer localement les bénéfices auxquels il peut prétendre, basé sur le formulaire complètement rempli; ③ il exécute un processus d'*Exposition Minimum* pour calculer un ensemble minimum d'assertions utiles (i.e. de données à fournir) dans le formulaire pour obtenir le service demandé avec les bénéfices souhaités; ④ il valide et signe le formulaire contenant cet ensemble minimum d'assertions et l'envoie au Fournisseur de Service. Ce dernier peut ⑤ exécuter son processus de décision en utilisant les données contenues dans le formulaire et calibrer ainsi l'offre de service proposé à l'Utilisateur. La déclaration du candidat est ensuite stockée par le Fournisseur de Service pour une durée qui dépend du contexte. A n'importe quel moment après l'étape ④ le candidat peut être amené à fournir les documents certifiant qu'il est bien capable de prouver les assertions énoncées dans le formulaire envoyé. Notez que cette phase de vérification n'est pas obligatoire, ou peut être réalisée pour un sous ensemble de valeurs du formulaire (e.g. une déclaration médicale liée à une assurance n'est souvent vérifiée que lors d'une demande de prime).

2.2. Contexte

2.2.1. Règles de Collecte

Les Règles de Collecte (*RC*) décrivent les informations requises par le Fournisseur de Service et les bénéfices associés.

En général, les processus de prise de décision sont basés sur des mécanismes de *boîte blanche*, c'est à dire compréhensible et justifiable par un humain, et donc public. C'est le cas pour les applications administratives (e.g. impôts, aides sociales diverses où le processus est documenté), mais aussi pour les applications commerciales respectueuses du droit. En effet, comme montré dans (Huysmans *et al.* 2007), une telle contrainte est imposée par la loi et/ou pour une meilleure acceptation par les utilisateurs de processus dans de nombreux domaines. Par exemple, la loi américaine "Equal Credit Opportunity Act" impose une transparence dans les règles de prise de décision concernant les prêts immobiliers[13]. Il en est de même pour de nombreux systèmes médicaux. Certaines études récentes comme (Baesens, *et al.,* 2003) transforment des mécanismes de prise de décision de type "boîte noire" comme les SVM ou les réseaux de neurones, en des mécanismes de type "boîte blanche".

Dans certains cas, la logique métier derrière le processus de prise de décision doit rester secrète. Dans ce cas, les règles de collection ne peuvent pas être dévoilées aux utilisateurs. Pour adresser ce cas de règles de collection privées, nous introduisons un *matériel sécurisé* côté utilisateur, qui peut par exemple être

[13] Voir http://www.ftc.gov/bcp/edu/pubs/consumer/credit/cre15.shtm

configuré en partie par le Fournisseur de Service. Ce matériel sécurisé contient les formulaires, les règles de collection, le sélectionneur de bénéfices et un algorithme d'*Exposition Minimum*. Nous requérons de la part de ce matériel qu'il soit sécurisé vis-à-vis de l'Utilisateur. Un bon candidat serait une carte à puce basique (e.g. utilisant le contrôleur sécurisé STM32-Discovery) capable d'exécuter un algorithme de minimalisation. Comme ces microprocesseurs ne sont pas très puissants, et ne disposent que de peu de RAM, ils ne peuvent exécuter que les algorithmes les plus simples (RAND*). Nous avons démontré dans (Anciaux *et al.*, 2013) la faisabilité de l'approche et la qualité de la minimalisation en utilisant un tel matériel.

De surcroît, certaines règles de décision peuvent être en pratique très compliquées : des prêts sont accordés suite à l'exécution d'arbres de décision, de SVM ou de réseaux de neurones (Crook *et al.*, 2007). Les règles de collection doivent donc avoir un pouvoir expressif suffisant pour capturer le processus de prise de décision du Fournisseur de Service. Dans ce papier, nous considérons des ensembles de règles de collection, chacune modélisée par un ensemble de disjonctions de conjonctions de contraintes sur des paires attribut/valeur. Ce modèle est facile à comprendre ; il est également très expressif, puisqu'il couvre les arbres de décision, très utilisés en pratique (Mitchell, 1997), ainsi que les forêt d'arbres de décision, utilisés pour des décisions multi-dimentionnelles. Par exemple, une banque proposant des prêts pourrait inclure une règle pour favoriser certaines familles ou de jeunes étudiants en leur fournissant une partie du prêt à taux zéro (*PTZ*), décrit par :

PTZ: (*marié=vrai* $\wedge$ *enfants>0*) $\vee$ (*age<30* $\wedge$ *Edu='Univ'*)

Il est réaliste de faire l'hypothèse que personne ne peut être obligé à fournir une assertion s'il ne le souhaite pas. Ce faisant, la seule pénalité est de ne pas pouvoir obtenir certains bénéfices. Ainsi, toutes les règles doivent être *positives*, en ce sens qu'il est bénéfique de les déclancher. Ceci n'est pas une limitation du modèle, puisque de telles règles qui conduiraient à un déni de bénéfice (appelées *règles négatives*) peuvent être construites en intégrant la négation de la règle dans l'ensemble des règles de collection. Par exemple, si le *PTZ* défini ci-dessus n'est *pas* accordé aux personnes ayant un casier judiciaire (*casier_judi=oui* $\Rightarrow$ $\neg$*PTZ*), la règle peut être réécrite :

PTZ: (*marié=vrai* $\wedge$ *enfants>0* $\wedge$ *casier_judi=non*) $\vee$ (*age<30* $\wedge$ *Edu='Univ'* $\wedge$ *casier_judi=non*)

2.2.2. Assertions Utilisateur et Documents

La granularité à laquelle sont signées les données a un impact sur la résultat et la qualité du processus de minimisation. A l'heure actuelle, puisqu'il n'y a aucune raison de procéder autrement, les documents officiels sont transmis dans leur intégralité, c'est-à-dire qu'ils sont un agrégat d'informations. Un utilisateur n'aura donc comme choix que de montrer ou cacher l'intégralité des documents qu'il possède. Cependant, il n'y a aucune difficulté technique à produire des données atomiques de la forme (attribut, valeur), que nous appelons des *assertions*, un document constituant alors un ensemble de données atomiques. Dans ces conditions,

le processus de minimisation peut travailler sur ces paires de manière indépendante, et décider individuellement de les exposer ou non.

Nous distinguons ainsi trois types de documents (au sens large): des *assertions* qui sont utilisées pour remplir des *formulaires*, et des *documents (agrégats) officiels* signés par les Producteurs de Données, conservés par les utilisateurs, et utilisés pour prouver la validité de leurs assertions. Les *formulaires* sont signés par l'utilisateur et les *documents officiels* sont signés par des Producteurs de Données à des granularités variables. Par exemple, un utilisateur souhaite valider dans un formulaire de demande d'aide au logement les champs $form_{AL}$:{*nombre_enfants>0, salaire>30.000*}. Il possède comme document officiel son avis d'imposition, signé par le ministère des finances, qui contient (entre autres) les informations suivantes doc_{IMPOTS}:{*nombre_enfants*=2, *salaire*=45.248, *marié*=vrai, *age*=40, etc}. L'utilisateur peut utiliser ce document officiel pour produire et signer deux assertions as_1:*nombre_enfants*=2 et as_2:*salaire*=45.248 qui permettent de valider les deux champs du formulaire, qu'il signera également. En cas de contrôle, l'utilisateur devra fournir doc_{IMPOTS} qui est un document signé par une autorité prouvant les assertions qu'il a produites.

Comme dans la plupart des formulaires, une valeur exacte n'est pas demandée, mais plutôt un intervalle, nous considérons que les assertions sont de la forme (*attribut* θ *valeur*) avec $\theta \in \{<, \leq, =, \neq, \geq, >\}$, ce qui conduit à exposer moins d'informations que d'utiliser le comparateur d'égalité. Dans ce qui suit, pour ne pas compliquer inutilement les discussions, nous considérons également que les documents officiels sont des ensembles de cardinalité 1, ce qui fait que chaque assertion *as* peut être prouvée par un document officiel *doc* sans fuiter plus d'informations que le contenu d'*as*. Nous discutons en Section 3.3 des implications dans les cas où les documents officiels contiennent plusieurs éléments.

2.2.3. Métriques d'estimation du degré d'exposition

La minimisation des données transmises au Fournisseur de Service, suite à l'application d'une technique d'Exposition Minimum, produit un double bénéfice, à la fois pour l'utilisateur et pour le fournisseur : d'une part la réduction du nombre d'assertions exposées est appréciable (de manière assez triviale) pour l'utilisateur du point de vue la protection de sa vie privée, puisqu'il voit ses données personnelles moins diffusées. D'autre part, la réduction des données transmises se traduit par une réductions des coûts pour le Fournisseur de Service, que ce soit un coût de traitement, qui est proportionnel à la taille des données à traiter ou vérifier, ou un risque financier dans le cas d'une fuite d'information, selon les facteurs dominants suivants, qui sont définis dans (Ponemon, 2011) : a) la réponse ex-post, qui représente 20% du coût d'une fuite de données personnelles. Elle inclue les actions prises par l'entreprise victime de la fuite pour venir en aide aux personnes dont elle a perdu les données en cherchant à minimiser la conséquence de cette perte. Plus il y a de données exposées, plus ce coût sera important. b) la perte de clientèle qui représente environ 50% du coût, qui est une conséquence directe de la médiatisation de l'affaire. Plus il y a de données exposées, plus l'opinion sera défavorable.

Tous ces aspects sont clairement liés à la quantité de données traitées. Dans cet article, nous construisons une métrique simple, *ad hoc* qui capture cette propriété. Toute métrique vérifiant la propriété d'indépendance de l'exposition pour plusieurs attributs différent peut convenir également, ce qui est le cas des métriques classiques basée sur l'entropie, comme *minimal distortion* (Samarati, 2001 ; Sweeney, 2002) ou *ILoss* (Xiao et Tao, 2006).

2.3. Exemple

Nous introduisons ici un scénario de prêt (voir Table 1) que nous utilisons comme fil rouge tout au long de l'article. Cet exemple est une simplification outrancière des applications réelles, puisque les systèmes d'aide à la décision dans le cadre des prêts bancaires, d'assurances médicales, ou de certaines aides sociales[14].

Une institution bancaire propose à tout le monde des prêts personnels de $5.000 à un taux de 10%, d'une durée de 1 an et avec une assurance de $50 par mois pour couvrir le risque de la perte d'emploi. Un prêt d'un montant plus important ($10.000) est proposé aux clients plus aisés, définis par la règle suivante :

$$(revenus > \$30.000 \wedge patrimoine > \$100.000) \\ \vee (nantissement > \$50.000 \wedge assurance\_vie = oui)$$

Cette règle est représentée par la Règle de Collection r_1 de la Table 1. La Règle de Collection r_2 exprime le fait qu'un prêt à un taux réduit peut être donné aux familles et à des jeunes de catégories peu risquées. La règle r_3 exprime le fait que de prêts sont accordés sur une durée plus longue pour des familles à haut revenus ou aux personnes de catégories peu risquées. La règle r_4 exprime le fait que l'assurance sur la perte d'emploi peut être réduite pour les familles riches ou les jeunes travailleurs prometteurs. Toutes ces règles de collection sont données dans la Table 1 sous la forme de disjonctions de conjonctions (forme DNF) de prédicats p_i de la forme *attribut* θ *valeur*.

Un utilisateur peut ainsi affirmer qu'il est *marié*, qu'il a 25 ans, qu'il a un enfant, un revenu annuel de $35.000, un diplôme universitaire, $75.000 de nantissement, un taux d'imposition de 11.5%, une assurance vie, et qu'il a reçu $250 de frais de sinistre l'année passée. Cette information est représentée sous la forme d'un ensemble d'assertions *attribut = valeur* notées as_1 à as_{10} dans la Table 1. Notez que ces assertions vérifient $\forall\, i,\ as_i \Rightarrow p_i$. Cet utilisateur peut ainsi déclancher l'ensemble des bénéfices c_1, c_2, c_3, c_4. Le processus d'Exposition Minimum doit maintenant identifier un ensemble minimum d'assertions permettant de déclancher ce même ensemble.

[14] Dans le cadre de notre collaboration avec le Conseil Général des Yvelines, nous avons travaillé sur le GEVA (voir http://www.mdph2b.fr/fichiers/1267003887-manuel_geva_mai_2008-2.pdf) qui présente plusieurs centaines de champs.

3. Le problème de l'Exposition Minimum

Cette section définit formellement le problème de l'Exposition Minimum sous deux formes, avec et sans poids, et étudie sa complexité.

3.1. Formalisation du problème

Nous notons $|S|$ la cardinalité d'un ensemble S. Nous introduisons ci-dessous les autres définitions requises, puis nous exprimons le problème. Nous illustrons chacune des notions introduites par des exemples de la Table 1.

Table 1. Règles de Collecte, Formulaire et Assertions pour le scénario de prêt

Règles de Collecte :

r_1: $(p_1 \wedge p_2) \vee (p_3 \wedge p_4) \Rightarrow c_1$
r_2: $(p_5 \wedge p_6 \wedge p_7) \vee (p_4 \wedge p_8 \wedge p_9) \Rightarrow c_2$
r_3: $(p_1 \wedge p_6 \wedge p_7) \vee (p_2 \wedge p_4 \wedge p_{10}) \Rightarrow c_3$
r_4: $(p_2 \wedge p_5 \wedge p_6 \wedge p_7) \vee (p_1 \wedge p_4 \wedge p_8 \wedge p_9) \Rightarrow c_4$

Avec

p_1: revenu_annuel>$30.000, p_2: patrimoine>$100.000,
p_3: nantissement>$100.000, p_4: assur_vie='oui',
p_5: taux_impots>10%, p_6: marié=vrai,
p_7: enfants>0, p_8: edu='université',
p_9: age<30, p_{10}: frais_sinistres<$5.000.

Et

c_1=prêt_élevé, c_2=taux_5%,
c_3=prêt_long, c_4=assurance_réduite.

***Formulaire* :**

revenu_annuel?, patrimoine?, nantissement?, assurance_vie?, taux_impots?, marié?, enfants?, edu?, age?, frais_sinistres?

Assertions Utilisateur ($data_u$):

as_1: revenu_annuel=$35.000, as_2: patrimoine=$150.000,
as_3: nantissement=$75.000, as_4: assur_vie='oui',
as_5: taux_impots=11.5%, as_6: marié=vrai,
as_7: enfants=1, as_8: edu='univ',
as_9: age=25, as_{10}: frais_sinistres=$250.

3.1.1. Définitions et notations

Attributs. Soit $A = \{a_i\}$ un ensemble fini d'attributs. A chaque attribut a_i on associe un domaine $dom(a_i)$.

Classes. Soit $C = \{c_j\}$ un ensemble fini de variables Booléennes, qu'on interprète comme des classes *positives* auxquelles un utilisateur peut appartenir. Si c_j=*vrai* pour un utilisateur donné, cela signifie qu'il peut obtenir le bénéfice associé à c_j.

Predicats. Nous appelons *prédicat sur A* toute expression de la forme $a\theta v$ where $a \in A$, $v \in dom(a)$ and $\theta \in \{=, <, >, \leq, \geq, \neq\}$.

Exemple: p_1: *revenu_annuel>$30.000* est un prédicat.

Assertions. Soit as_i une assertion composé d'un prédicat unique (en général d'égalité) sur *A*. Nous notons $data_u=\{as_i\}$ l'ensemble des assertions qu'un utilisateur donné *u* peut affirmer de manière vraie (i.e. il possède un document signé qui prouve cette assertion). Nous disons qu'une assertion as_i *prouve* un prédicat *p* si $as_i \Rightarrow p$.

Règles Atomiques. Une règle atomique menant à une classe c_j, notée $atom_j$ est une conjonction de prédicats tels que $atom_j=vrai \Rightarrow c_j=vrai$. Puisqu'il y a en général plusieurs règles atomiques qui mènent à une classe c_j nous les notons $atom_{j,k}$ en utilisant *k* pour les différencier.

Exemple: $atom_{1,1}$: (*revenu_annuel>$30.000* $\wedge$ *patrimoine>$100.000*) and $atom_{1,2}$: (*nantissement>$50.000* $\wedge$ *assur_vie='oui'*) sont deux règles atomiques menant à la classe c_1.

Table 2. Formule Booléenne de l'Ensemble des Règles du scénario du prêt

$B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$ t.q. $\forall i \in [1;10]$, $b_i=vrai \Leftrightarrow as_i$ *est exposé.*

La Formule Booléenne de l'Ensemble de Règles E_R est la suivante :

$E_R = ((b_1 \wedge b_2) \vee (b_3 \wedge b_4))$
$\wedge ((b_5 \wedge b_6 \wedge b_7) \vee (b_4 \wedge b_8 \wedge b_9))$
$\wedge ((b_1 \wedge b_6 \wedge b_7) \vee (b_2 \wedge b_4 \wedge b_{10}))$
$\wedge ((b_2 \wedge b_5 \wedge b_6 \wedge b_7) \vee (b_1 \wedge b_4 \wedge b_8 \wedge b_9))$

Supposons qu'un utilisateur ne peut affirmer que les assertions 1 à 9, la Formule Booléenne se construit alors en supprimant toutes les classes et règles atomiques qui ne peuvent pas être prouvées.

$E_R = ((b_1 \wedge b_2) \vee (b_3 \wedge b_4))$
$\wedge ((b_5 \wedge b_6 \wedge b_7) \vee (b_4 \wedge b_8 \wedge b_9))$
$\wedge ((b_1 \wedge b_6 \wedge b_7))$
$\wedge ((b_2 \wedge b_5 \wedge b_6 \wedge b_7) \vee (b_1 \wedge b_4 \wedge b_8 \wedge b_9))$

Nous disons qu'un ensemble d'assertions $Data_u=\{as_i\}$ prouve une règle atomique $atom_{j,k} = \wedge_m \ q_{j,k,m}$ où $q_{j,k,m}$ est un prédicat sur *A*, si et seulement si $\forall j,k,m \ \exists \ i : as_i \Rightarrow q_{j,k,m}$. On dit que $data_u$ *prouve* $atom_{j,k}$ *de manière unique* si et seulement si $\forall j,k,m \ \exists! i: as_i \Rightarrow q_{j,k,m}$.

Exemple: $data_u=\{as_1, as_2, as_3, as_3\}$ prouve de manière unique les règles atomiques $atom_{1,1}$ et $atom_{1,2}$.

Règles de Collection. Une règle de collection r_j est une disjonction de règles atomiques menant à la classe c_j. Formellement: $r_j: \vee_k \ atom_{j,k}$. Si un ensemble d'assertions $data_u$ prouve une règle atomique $atom_{j,k}$ alors on dit que $data_u$ *prouve* r_j, ce qui signifie que l'utilisateur *u* peut obtenir les avantages associés à c_j (de manière triviale, $r_j=vrai \Rightarrow c_j=vrai$).

Exemple: r_1: (*revenu_annuel>$30.000* $\wedge$ *patrimoine>$100.000*) $\vee$ (*nantissement>$50.000* $\wedge$ *assurance_vie='oui'*) est une règle de collection qui mène à la classe *prêt_élevé*.

Dans ce qui suit, nous écrivons $r_j = \vee_k (\wedge_m \ q_{j,k,m})$ où $q_{j,k,m}$ est un prédicat sur A. Considérons r_1 dans l'exemple précédent, nous avons $q_{1,1,1}$: *revenu_annuel>$30.000*, $q_{1,1,2}$*: patrimoine>$100.000*, $q_{1,2,1}$*:nantissement>$50.000* et $q_{1,2,2}$*: assur_vie=oui*.

Ensemble de Règles. Soit $R = \{r_j\}$ un ensemble de $|C|$ *règles de collection*, une par classe c_j. Si $data_u$ prouve (resp. de manière unique) toutes les règles de R alors on dit que data_u prouve (resp. de manière unique) R.

Formule Booléenne d'un Ensemble de Règles. Si une seule assertion prouve de manière unique un prédicat donné dans les règles, décider si data_u prouve l'ensemble de règles R est *équivalent* à tester la valeur de vérité d'une formule Booléenne $E_R=\wedge_j(\vee_k(\wedge_m b_{f(j,k,m)}))$ appelée *Formule Booléenne de l'Ensemble de Règles R*, où *f(j,k,m)* est une fonction de domaine [1;|A|] définie par $f(j,k,m)=i \Leftrightarrow (as_i \Rightarrow q_{j,k,m})$ et $b_{f(j,k,m)}$ est une variable Booléenne qui vaut *vrai* si $as_{f(j,k,m)}$ est exposé et *faux* dans le cas contraire. Notons que si on considère l'affectation qui met toutes les valeurs de $b_{f(j,k,m)}$ à *vrai*, alors E_R=vrai $\Leftrightarrow$ $data_u$ *prouve R*.

Exemple: La Table 2 illustre la Formule Booléenne de l'Ensemble de Règles R de la Table 1.

Métrique d'Exposition. Soit $B=\{b_i\}$ un ensemble de variables Booléennes. Soit T_B une affectation de ces variables telle que $b_i=vrai \Leftrightarrow as_i$ est envoyé au fournisseur de service. On note $\mathbf{EX}(T_B)$ une fonction représentant l'exposition de l'ensemble des assertions exposées. L'exposition est proportionnelle au coût financier pour le fournisseur de service, et à la perte d'informations personnelles pour l'utilisateur.

Table 3. Notations Algorithmiques

$D=|Data_u|=10$; $C=4$;
B[] est un tableau de Booléens de taille D tel que:
$\forall i \in [1;10]$, $B[i]=vrai \Leftrightarrow as_i$ est *exposé*
R[] est un tableau de C règles de collection;
R[j].atom[] pour $j \in [1;4]$ sont 4 tableaux de 2 règles atomiques (car il y a 1 seule disjonction);
R[j].atom[k].b[] pour $j \in [1;4]$, $k \in [1;2]$ sont des tableaux de références vers les éléments de *B[i]*. On note **B[i]*une référence vers *B[i]*.
Les *R[j].atom[k].b[m]* sont définis ainsi :
R[1].atom[1].b[1]←**B[1]*; *R[1].atom[1].b[2]*←**B[2]*; (…)
R[2].atom[2].b[2]←**B[8]*; *R[2].atom[2].b[3]*←**B[9]*; (…) *R[4].atom[2].b[3]*←**B[8]*;
R[4].atom[2].b[4]←**B[9]*;

Exemple: La fonction $\mathbf{EX_1}(T_B) = |\{ b_x \in B: T_B(b_x)=vrai \}|$ qui compte le nombre d'assertions publiées peut être utilisée comme une métrique d'exposition. Notons que toute métrique invariante au cours du temps, étant donné une affectation T_B peut être utilisée. En particulier, ceci inclut les métriques de type *information loss* qui peuvent raisonnablement être supposées proportionnelles à $\mathbf{EX_1}$. Désormais, si une affirmation est publiée, nous disons qu'elle est *exposée*.

3.1.2. Enoncé du problème d'Exposition Minimum Booléen

Nous pouvons maintenant définir le problème de décision de l'Exposition Minimum d'un ensemble d'assertions $data_u$ par rapport à un ensemble de règles R et une métrique d'exposition **EX**. Notons que nous supposons que $data_u$ prouve R. Si ce n'était pas le cas, nous considèrerions simplement R' le sous ensemble de règles de R prouvé par $data_u$. L'objectif est de trouver une affection T_B des variables Booléennes associées à la publication des assertions qui minimise leur exposition, calculée en utilisant la métrique **EX**.

Problème de décision *n-exposition*:

Etant donné un ensemble de règles R, $data_u=\{as_x\}$ un ensemble de q assertions qui prouvent R **de manière unique**, $B=\{b_1,..,b_q\}$ un ensemble de variables Booléennes tel que $b_x=vrai \Leftrightarrow as_x$ est exposé, $E_R=\wedge_j(\vee_k(\wedge_m b_{j,k,m}))$ où $\forall j,k,m\ b_{j,k,m}\in B$ la Formule Booléenne de l'Ensemble de Règles R, et la fonction d'exposition **EX**, alors $data_u$ est *n-exposable* par rapport à R si et seulement si il existe une affectation T_B de B telle que $\mathbf{EX}(T_B)\leq n$ et E_R est *vraie*.

Exemple: Considérons R et $data_u$ définis à la Table 1. Il est évident que $data_u$ prouve R. De plus, $data_u$ est 5-exposable par rapport à 5 et la métrique $\mathbf{EX_1}$, puisque l'affectation $T_B=\{b_1=T,\ b_2=T,\ b_3=F,\ b_4=F,\ b_5=T,\ b_6=T,\ b_7=T,\ b_8=F,\ b_9=F,\ b_{10}=F\}$ satisfait E_R et $\mathbf{EX_1}(T_B)=5$.

Nous nous intéressons plus particulièrement au problème d'optimisation associé, appelé *problème d'Exposition Minimum* dont le but, étant donné $data_u$, R et **EX** est de trouver la valeur minimale de n de telle sorte que $data_u$ soit n-exposable.

3.2. Analyse de Complexité

3.2.1. Analyse de Complexité

Nous analysons ici la complexité et l'approximabilité du problème *EM*. Nos résultats concernent à la fois des schémas d'approximation en temps polynomial (Papadimitriou et Yannakakis, 1988), et également l'approximation différentielle (Escoffier et Paschos, 2007)[15].

THEOREM 1.
Le problème de décision (resp. de minimisation) All Positive Min Weighted SAT (MinSAT pondéré à valeurs positives) est réductible au problème de décision de n-exposition (resp.au problème d'optimization d'Exposition Minimum).

[15] Etant donné une instance I d'un problème d'optimisation et une solution S de I, on note $m(I,S)$ la valeur de la solution S, $opt(I)$ la valeur d'une solution optimale de I et $W(I)$ la valeur d'une pire solution de I. Le facteur approximation différentielle de S est définie par $DR(I,S)=abs((m(I,S)-W(S))/(opt(I)-W(I))$. Le facteur classique d'approximation pour un problème de minimisation est simplement défini par $m(I,S)/opt(I)$.

PREUVE. Le problème de décision (resp. d'optimisation) *Min Weighted Sat* decision (resp. optimization) est défini dans (Alimonti *et al.*, 1998) de la manière suivante :

"Etant donné un entier n, une instance $\{P_{j,k}\}$ de P variables Booléenne, une formule en CNF $F=\wedge_j (\vee_k P_{j,k}))$ sur $\{P_{j,k}\}$ et une fonction (positive) de poids $w:\{P_{j,k}\}\rightarrow R+$, trouver une affectation T des $\{P_{j,k}\}$ qui satisfait F telle que $w(T)= \Sigma_{j,k} w(P_{j,k})\times T(P_{j,k})$ est $< n$ (resp. est minimum)."

Lorsqu'une formule ne contient aucune variable avec négation, le problème est appelé *All Positive Min Weighted Sat* (*APMWS*). Le problème de décision de n-exposition (resp. problème d'optimization *EM*) considère une formule $E=\wedge_j (\vee_k (\wedge_m b_{j,k,m}))$. Une instance d'un problème *APMWS* peut être mappé à un problème de n-exposition (resp. d'optimization *EM*) en choisissant $\forall j,k$: $m=1$ et $b_{j,k,1}=P_{j,k}$ (c-à-d, toutes les règles atomiques ne contiennent qu'un seul prédicat). Toute solution du problème de n-exposition (resp. d'optimization *EM*) c-à-d., trouver une affectation de poids minimum de $b_{j,k,1}$ telle que E_R est *vraie* sera une solution du problème *APMWS* en choisissant $P_{j,k}=b_{j,k,1}$■

COROLLAIRE 1.
Le problème de n-exposition est NP-Complet.

PREUVE. Puisque le problème *APMWS* est NP-Complet, et étant donné le THEOREME 1 le résultat est immédiat. ■

COROLLAIRE 2.
Le problème d'optimisation EM est NP-Difficile, n'est pas dans APX[16]*, et a un facteur d'approximation différentiel de 0-DAPX*[17].

PREUVE. Les résultats de complexité sont une conséquence directe du THEOREME 1, et de résultats de complexité négatifs du problème *APMWS*. Le Théorème 6 d'Alimonti *et al.* (1998) énonce que le problème *APMWS* n'est pas dans APX, ce qui montre que *EM* n'y est pas non plus. Escoffier et Paschos (2007) étudient le problème *APMWS* du point de vue de l'approximation différentielle, et montrent que le problème est de la classe 0-DAPX. Ainsi le problème d'optimisation *EM* est aussi dans 0-DAPX. ■

3.2.2. Résolution du problème EM

Le COROLLAIRE 2 est un résultat négatif en terme de complexité qui montre que le problème est difficile, et que les schémas d'approximation en temps polynomial ne vont fournir aucune garantie en terme de proximité de la solution avec l'optimum. Les résultats expérimentaux (voir Section 5) sont cependant encourageants.

[16] APX est la classe des problème d'optimisation NP qui admettent un schéma d'approximation en temps polynomial avec un facteur d'approximation borné par une constante.
[17] 0-DAPX est la classe des problèmes d'optimisation NP pour lesquels toutes les approximation polynomiales ont un facteur d'approximation différentiel de 0.

3.3. Généralisation du problème

3.3.1. Définition du problème

Nous avons considéré à la Section 3.1. le cas où les Producteurs de Données étaient capables de fournir des documents composés d'un seul prédicat *attribut=valeur*, avec une signature à cette granularité également. Dans un cadre plus général, où cette hypothèse ne tient plus, le problème est que l'ensemble des assertions de l'utilisateur ne prouve plus *de manière unique* l'ensemble des règles. Ce problème peut être exprimé de la manière suivante, en utilisant une métrique $\mathbf{EX_{DATA}}$ qui s'applique à un ensemble d'assertions au lieu d'un ensemble de variables Booléennes :

Problème de *n-exposition* généralisé
Etant donné un ensemble de règles R, $data_u=\{as_i\}$ un ensemble d'assertions prouvant R, et une fonction d'exposition $\mathbf{EX_{DATA}}$. On dit que $data_u$ est *n-exposable* par rapport à R si on peut trouver $data_{min} \subset data_u$ tel que $data_{min}$ prouve R et $\mathbf{EX_{DATA}}(data_{min}) \leq n$.

Dans ce cadre, chaque prédicat d'une règle peut potentiellement être prouvé par plusieurs assertions (par exemple, un utilisateur disposant de plusieurs documents officiels pourrait être en mesure de produire les deux assertions suivantes : *salaire > $1.000* et *salaire > $3.000*). Notons que ce problème couvre également le cas où les documents officiels utilisés pour prouver les assertions ne sont pas atomiques : dans ce cas plusieurs documents officiels peuvent prouver la même assertion, et donc le même prédicat des règles.

3.3.2. Résolution du problème

Dans le cas général, il n'y a plus équivalence entre la Formule Booléenne de l'Ensemble des Règles et le calcul d'une solution du problème d'Exposition Minimum. Il faut en réalité calculer une formule plus compliquée, notée E'_R dans laquelle chaque prédicat de chaque règle est remplacé par la disjonction des assertions pouvant le prouver.

Problème de *n*-exposition Multi-Preuves
Etant donné un ensemble de règles R, $data_u=\{as_i\}$ un ensemble de q assertions qui prouve R, $B=\{b_1,..,b_q\}$ un ensemble de variables Booléennes tel que $b_i=vrai \Leftrightarrow as_i$ est exposé, $E'_R=\wedge_j(\vee_k(\wedge_m(\vee_q b_{j,k,m,t})))$ où $\forall j,k,m,t\ b_{j,k,m,t} \in B$, et la fonction d'exposition $\mathbf{EX_{MULTI}}$, $data_u$ est *n-exposable* par rapport à R si et seulement si il existe une affectation T_B de B telle que $\mathbf{EX_{MULTI}}(T_B) \leq n$ et E'_R est *vraie*.

Dans ce qui suit, on considère le problème d'optimisation associé *Exposition Minimum Multi-Preuves* (EM2P). Il est évident que le problème *EM* est réductible au problème *EM2P*, puisque c'est un cas particulier pour lequel $\forall j,k,m\ t=1$. Ainsi les résultats de complexité des Corrolaires 1 et 2 s'appliquent toujours.

La problématique nouvelle introduite par la multi-prouvabilité conduit à l'adaptation de la fonction d'exposition pour prendre en compte le fait que toutes les expositions ne sont pas comparables. Par exemple l'exposition de l'assertion *salaire > $5.000* est moindre que l'exposition de l'assertion *salaire > $6.000*. On introduit

ainsi $\mathbf{EX_{PRED}}$, une fonction calculant l'exposition d'un prédicat p de la forme $a\ \theta\ v$, où $dom(a)$ est fini par :

si $\theta \in \{=\}$: $\mathbf{EX_{PRED}}(p) = 1$
si $\theta \in \{<, >, \leq, \geq, \neq\}$: $\mathbf{EX_{PRED}}(a\theta v) = 1 - |\{x \in dom(a): x\theta v\}| / |dom(a)|$

On introduit également une fonction $\mathbf{EX_{MULTI}}$ qui calcule l'exposition d'un ensemble de prédicats sur un ensemble d'attributs exposés A en sommant la valeur maximale de leur exposition, calculée par $\mathbf{EX_{PRED}}$:

$$\mathbf{EX_{MULTI}}(B) = \Sigma_{a \in \text{attributs publiézs}} \text{MAX}_{p = a\theta v}(\mathbf{EX_{PRED}}(p))$$

En utilisant $\mathbf{EX_{MULTI}}$ comme fonction objectif et en choisissant pour chaque p_i l'assertion $as_{j,k,m}$ le prouvant et minimisant $\mathbf{EX_{MULTI}}$, alors chaque prédicat de l'ensemble des règles peut être remplacé par l'*unique* assertion le prouvant *et* minimisant $\mathbf{EX_{MULTI}}$, et ainsi résoudre le problème *EM2P* revient à résoudre le problème *EM*. Toutes les techniques de résolutions peuvent être adaptées, tout simplement en modifiant la fonction d'exposition.

4. Résolution Exacte et Algorithmes Approchés pour le Problème d'Exposition Minimum

Dans cette Section, nous donnons des méthodes exactes et approchées pour calculer une solution d'une instance d'un problème *EM*. Pour la résolution exacte, nous traduisons l'instance sous la forme d'un programme *AMPL*, que nous résolvons en utilisant un solver de Programmation Entière Binaire (*BIP*). Pour la résolution approchée, nous proposons plusieurs algorithmes polynomiaux : une approche aléatoire naïve, un algorithme de recuit simulé, et une heuristique spécifique, appelée *Algorithme EM*.

Dans tous ces algorithmes, nous considérons une formule Booléenne E_R construite comme indiqué dans la Section 3.1. en utilisant un ensemble de règles R composé d'un ensemble de C règles de collection associé à des bénéfices (ou classes) que l'utilisateur peut (et souhaite) obtenir. On suppose que chaque règle atomique peut être prouvée en utilisant des assertions basées sur des documents officiels possédés par l'utilisateur, et si ce n'est pas le cas, on retire de R les règles et bénéfices qui ne peuvent pas être déclenchés ou que l'utilisateur ne souhaite pas obtenir en utilisant le *Sélectionneur de Bénéfices* (voir description de la Figure 1, étape ②). La taille de $data_u$, l'ensemble des assertions de l'utilisateur est notée D.

T_B est une fonction d'affectation de $data_u$ que nous implémentons comme un tableau B de Booléens avec la sémantique suivante : $B[i]=vrai \Leftrightarrow as_i$ *est exposé*. L'ensemble des règles est représenté par un table $R[]$ de règles de collection, chaque règle $R[i]$ est un tableau $atom[]$ de règles atomiques, et chaque règle atomique $R[i].atom[j]$ est un tableau $b[]$ de références vers les éléments de B (voir exemple en Table 3). Notons que E_R est *vraie* lorsque chaque règle de collection $R[i]$ possède au moins une règle atomique où tous les éléments Booléens référencés sont *vrais*.

4.1. Résolution Exacte (Modèle BIP)

Traditionnellement on peut utiliser un *solver* BIP, ou plus généralement un solver de type *MINLP* (Programmation Entière Mixte Non-Linéaire), pour obtenir l'optimum d'un problème d'optimisation, représenté sous forme de Programme (ici non linéaire sur des variables binaires). Nous avons choisi pour calculer la solution du problème un solver très utilisé, *COUENNE* (voir Belotti *et al.*, 2009).

La représentation du problème est indépendante du solver utilisé, et doit être écrite comme un programme MINLP. L'écriture d'un programme *non-linéaire* est très simple compte tenu de la modélisation choisie, puisqu'il s'agit d'une transformation directe où chaque document correspond à une variable Booléenne, et où la fonction objectif est la somme de toutes les variables binaires, pouvant prendre les valeurs 0 ou 1. Pour chaque règle de collection r_j, on exprime une contrainte non-linaire : r_j: $\Sigma_k \Pi_m as_{j,k,m} \geqslant 1$.

Il existe plusieurs langages pour écrire un tel programme. Nous avons choisi le populaire *AMPL* (voir Fourer *et al.*, 1990) L'exemple présenté en Section 2.3 se représente de la manière suivante en *AMPL* :

```
var b1 binary; var b2 binary; ... var b10 binary;
minimize EX:
b1+b2+b3+b4+b5+b6+b7+b8+b9+b10;
subject to
r1: b1*b2 + b3*b4 >= 1;
r2: b5*b6*b7 + b4*b8*b9 >= 1;
r3: b1*b6*b7 + b2*b4*b10 >= 1;
r4: b2*b5*b6*b7 + b1*b4*b8*b9 >= 1;
```

Ce programme est ensuite traité par le solver. Comme nous le montrons dans la Section 5, la zone de paramètres pour lesquels le solver fournit une réponse dans un temps raisonnable (moins de 10 minutes) est petite. Notons que nous travaillons actuellement à la linéarisation de ce problème, ce qui permettra peut être d'étendre le domaine où le solver produit un résultat, car les problèmes d'optimisation non-linéaires sont *très difficiles*.

4.2. Solutions Approchées (Temps Polynomial)

Pour les instances de problème ne pouvant être résolues de manière exacte, il est nécessaire de faire appel à un algorithme polynomial approché. Nous proposons trois algorithmes : un algorithme naïf aléatoire, appelé *RAND**, un algorithmes basé sur une méta-heuristique (nous avons choisi le recuit simulé), appelé *SA** et un algorithme exploitant une heuristique particulière de notre problème *EM*, appelé *HME*. Ces algorithmes sont non déterministes, ainsi ils peuvent être exécutés plusieurs fois, tout en conservant le meilleur résultat. Ils produisent leur premier résultat en temps linéaire ou polynomial, selon l'algorithme. Nous discutons leur complexité sur *une seule exécution*, afin de comparer leur vitesse. Pour comparer leur efficacité, nous exécutons l'algorithme le plus lent (*HME* en l'occurence) une

seule fois, et nous exécutons pendant la même durée les autres algorithmes (*SA** et *RAND**) autant de fois que possible, tout en conservant leur résultat optimum.

4.2.1. Algorithmes Polynomiaux

Nous utilisons deux algorithmes polynomiaux, *RAND** et *SA** que nous détaillons plus précisément dans (Anciaux *et al.*, 2012). Ces algorithmes servent essentiellement de borne inférieure en terme de qualité du résultat. *RAND** calcule une solution possible aléatoire, et peut être lancé autant de fois que nécessaire (tant qu'il reste du temps processeur disponible). *SA** est une implémentation de la méta-heuristique de recuit simulé, où nous avons adapté la métrique de voisinage à notre problème, une solution initiale à affiner par la méta-heuristique étant donnée par une exécution de *RAND**. Ces deux algorithmes sont de complexité polynomiale (linéaire).

4.2.2.Algorithme HME

L'algorithme *HME* (pour *Heuristic for Minimum Exposure*) est basé sur une heuristique propre au problème *EM*. Son code est donné en Table 5. Nous illustrons son fonctionnement sur un exemple, et discutons sa complexité. L'heuristique se trouve dans le calcul de la fonction *score[i]* à la ligne 6, qui calcule le score de la $i^{\text{ème}}$ variable Booléenne de *B* en utilisant la fonction *fix*(*B*). Cette fonction calcule une borne inférieure sur la valeur de $\mathbf{EX_1}$ en calculant le nombre de prédicats qui ne peuvent plus être mis à *faux* étant donné *B*. Par exemple, supposons que *B[i]=faux* (c'est-à-dire que l'assertion as_i n'est pas exposée). Toutes les règles atomiques qui se réfèrent à *B[i]* ne peuvent plus être prouvées. Ceci conduit au fait que $\mathbf{EX_1}$ sera supérieure ou égale à la valeur de la cardinalité de l'ensemble des prédicats des règles atomiques qui sont les *seules restantes* pour prouver un bénéfice. En nous basant sur l'exemple de la Section 2.3, nous illustrons dans la Table 6 comment la fonction *fix*(*B*) est calculée, pour chaque étape de l'algorithme.

Décrivons brièvement comment les cases *score[1]* et *score[3]* sont calculées, pour la première ligne de la Table 6. Si *B[1]=faux*, alors il faut prouver les règles *R[1]*, *R[3]*, *R[4]* en utilisant respectivement les règles atomiques *R[1].atom[2]*, *R[3].atom[2]*, *R[4].atom[1]*, ce qui signifie mettre à *vrai* les 7 Booléens suivants : *B[2]*, *B[3]*, *B[4]*, *B[5]*, *B[6]*, *B[7]*, *B[10]*. En conséquence, *score[1]=7*. De même, si *B[3]=faux*, il faut prouver *R[1]* en utilisant *R[1].atom[1]*, ce qui signifie mettre à *vrai* les 2 Booléens suivants : *B[1]*, *B[2]*, et par conséquence, *score[3]=2*. Nous indiquons en gris le score le plus faible, qui donne une affectation à *faux* dans les étapes suivantes, indiqué par le symbole –. Les assertions pour lesquelles le score est noté ∞ sont celles pour lesquelles l'affectation finale sera *vrai*. Le résultat final ici est : *B=[B[1]=vrai, B[2]=vrai, B[3]=faux, B[4]=faux, B[5]=vrai, B[6]=vrai, B[7]=vrai, B[8]=faux, B[9]=faux, B[10]=faux]* qui, il se trouve, est également la valeur minimale de $\mathbf{EX_1}$ sur cette instance du problème.

On voit que le coût de *HME* est proportionnel à $O(\text{COST}_{\text{FIX}} \times \text{D}^2)$, où COST_{FIX} est le coût pour calculer la fonction *fix*. Plus précisément dans notre implémentation,

$COST_{FIX} = O(C \times d_R \times d_{QD})$, où d_R est le nombre de règles atomiques par règle de collecte, et d_{QD} est le nombre de prédicats par règle atomique.

Table 5. Algorithme HME

Algorithme HME

Input: E_R une formule booléenne d'un ensemble de règles
Ouput: *B* une affectation qui prouve *R*

```
1.for i = 1 to D do
2.  B[i] ← true
3.endfor
4.while (exists i such that: B[i]=true and
        if B[i] is set to false then E_R(B) remains true) do
5.  for i = 1 to D do
6.    score[i] ← ∞
7.  endfor
8.  forall i such that B[i] = true do
9.    B[i] ← false
10.       if E_R (B)=true then // E_R (B) is true iff B proves R
11.             score[i] ← fix(B)
12.       endif
13.       B[i] ← true
14.   endforall
15.   m ← i such that score[i] is minimum
16.   B[m] ← false
17. endwhile
18. return B
```

Table 6. Calcul de fix(B) pour une exécution de HME

Etapes	*B[1]*	*B[2]*	*B[3]*	*B[4]*	*B[5]*	*B[6]*	*B[7]*	*B[8]*	*B[9]*	*B[10]*
1: score[i]	7	7	**2**	5	4	6	6	4	4	3
2: score [i]	∞	∞	–	5	5	6	6	5	5	**4**
3: score [i]	∞	∞	–	**5**	7	∞	∞	5	5	–
4: score [i]	∞	∞	–	–	∞	∞	∞	**5**	5	–
5: score [i]	∞	∞	–	–	∞	∞	∞	–	**5**	–
Final B[i]	*vrai*	*vrai*	*faux*	*faux*	*vrai*	*vrai*	*vrai*	*faux*	*faux*	*faux*

L'intuition soutenant l'heuristique est d'éliminer successivement les assertions qui nécessitent de garder le moins d'autres assertions parmi celles restant, de telle sorte que tous les bénéfices continuent à être obtenus. Cette heuristique est

particulièrement efficace lorsque le nombre de règles atomique par règle de collecte est faible. Notons que si ce nombre augmente, alors *HME* tend vers *RAND**.

Nous montrons en Section 5 que l'algorithme *HME* produit de très bons résultats en terme de qualité, tout en conservant une complexité acceptable. Nous montrons maintenant comment étendre l'algorithme *HME* à des assertions du type *attribut* θ *valeur*, avec $\theta \in \{<, \leq, =, \neq, \geq, >\}$, et donc où une assertion n'est pas sûr de prouver *de manière unique* un prédicat de l'ensemble des règles.

4.2.3. Extension au problème EM2P

Comme nous l'avons expliqué à la Section 3.3.2. l'extension au problème *EM2P* se fait simplement par une modification de la fonction d'exposition. Dans cette section, nous discutons uniquement des changements que cela induit sur l'algorithme *HME*. L'idée est de change la fonction *fix*( ) pour utiliser $\mathbf{EX_{PRED}}$ et $\mathbf{EX_{MULTI}}$. La fonction *fix* doit calculer la borne minimale de $\mathbf{EX_{MULTI}}$, dans le cas où une assertion n'est pas transmise. Considérons l'exemple suivant, avec trois prédicats différents sur *revenu_annuel*. Les nouveaux prédicats sont : p_{11}*: revenu_annuel > \$10.000*, p_{12}*: revenu_annuel > \$20.000* et p_{13}*: revenu_annuel > \$30.000*. On suppose que l'utilisateur dispose des assertions équivalentes as_{11}, as_{12} et as_{13}. Il est important de noter que (par ex.) as_{13} permet de prouver p_{11}, p_{12} et p_{13}. L'ensemble des règles est le suivant :

r_1: $(p_{11} \wedge p_2) \vee (p_3 \wedge p_4) \Rightarrow c_1$
r_2: $(p_5 \wedge p_6 \wedge p_7) \vee (p_4 \wedge p_8 \wedge p_9) \Rightarrow c_2$
r_3: $(p_{12} \wedge p_6 \wedge p_7) \vee (p_2 \wedge p_4 \wedge p_{10}) \Rightarrow c_3$
r_4: $(p_2 \wedge p_5 \wedge p_6 \wedge p_7) \vee (p_{13} \wedge p_4 \wedge p_8 \wedge p_9) \Rightarrow c_4$

Afin de calculer $\mathbf{EX_{PRED}}$, il faut travailler sur un domaine fermé. On suppose par exemple que *revenu_annuel* appartient au domaine [*0*; *100.000*]. Dans ces conditions, $\mathbf{EX_{PRED}}(p_{11}) = 0.3$, $\mathbf{EX_{PRED}}(p_{12}) = 0.2$ et $\mathbf{EX_{PRED}}(p_{13}) = 0.1$. Pour tous les autres prédicats p, $\mathbf{EX_{PRED}}(p)=1$. On notera également que puisque $p_{13} \Rightarrow p_{12}$ et $p_{13} \Rightarrow p_{11}$ mettre p_{1A} à *faux* tout en laissant p_{13} à *vrai* n'apportera aucune amélioration en terme d'exposition des données. En revanche, p_{13} peut être mis à *faux* tandis que p_{11} et p_{12} seraient mis à *vrai*, ce qui conduirait à une réduction d'exposition égale à $\mathbf{EX_{PRED}}(p_{11})$-$\mathbf{EX_{PRED}}(p_{12})=0.1$. En ce qui concerne la fonction *fix( )*, mettre p_{13} à *faux* ne fixe que les prédicats p_2, p_5, p_6, et p_7.

Table 7. Execution de HME sur un problème EM2P

Etapes	p_{11}	p_{12}	p_{13}	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
1:score	7	6	4	4.2	**1.1**	4.1	3.3	5.3	5.3	4	4	0
2:score	∞	6.1	4.1	∞	–	4.2	4.3	5.3	5.3	4.1	4.1	**3.2**
3:score	∞	∞	**4.2**	∞	–	4.2	4.3	∞	∞	4.2	4.2	–
4:score	∞	∞	–	∞	–	**4.2**	∞	∞	∞	4.2	4.2	–
5:score	∞	∞	–	∞	–	–	∞	∞	∞	**4.2**	4.2	–
6:score	∞	∞	–	∞	–	**–**	∞	∞	∞	–	**4.2**	–
7:B[*i*]	*vrai*	*vrai*	*faux*	*vrai*	*faux*	*faux*	*vrai*	*vrai*	*vrai*	*faux*	*faux*	*faux*

Dans la Table 7, nous montrons une exécution de l'algorithme *HME* qui calcule *fix*( ) pour chaque étape. Le résultat est T_B=[p_{11}:*vrai*, p_{12}:vrai, p_{13}:*faux*, p_2:*vrai*, p_3:*faux*, p_4:*faux*, p_5:*vrai*, p_6:*vrai*, p_7:*vrai*, p_8:*faux*, p_9:*faux*, p_{10}:*faux*], et $\mathbf{EX_{MULTI}}(T_B)$=*4.2*. Par rapport à l'exemple présenté en Table 6, la même valeur de vérité a été trouvée pour les prédicats p_2 à p_{10} et l'algorithme a également été capable de ne pas exposer p_{13} tout en exposant p_{12} et p_{11}, ainsi, l'exposition a été légèrement réduite : au lieu d'exposer *revenu_annuel=$35.000* l'utilisateur expose *revenu_annuel >$20.000*.

5. Expérimentation

Dans cette section, nous présentons une validation expérimentale de notre approche. Son objectif est double : (1) montrer que le gain avec la métrique d'exposition est significative, que l'on utilise la résolution exacte ou les algorithmes d'approximation; (2) montrer que l'approche passe à l'échelle, en montrant que les algorithmes d'approximation produisent des résultats acceptables, lorsque la résolution exacte est impossible. Nous commençons par présenter les paramètres de l'expérimentation, puis nous donnons les résultats. Tout le code des algorithmes, les données, le générateur de problèmes *AMPL* sont disponibles[18].

5.1. Dispositif Expérimental

L'expérimentation a été conduite sur une station de travail HP avec un CPU Intel cadencé à 3.1GHz et 8GB de RAM. Les programmes ont été développés en Java 1.6 (x64). Le solver *COUENNE* est exécuté sur la même machine. Nous considérons les algorithmes *RAND**, *SA** et *HME*.

La métrique utilisée est $\mathbf{EX_1}$ telle que définie à la Section 3.1.1. Dans les figures qui suivent, nous mesurons la réduction d'exposition obtenue en utilisant cette métrique, c'est-à-dire le pourcentage de réduction du nombre d'assertions exposées, par rapport à une technique classique de collection limitée côté serveur (i.e. où *tous* les documents impliqués dans les règles de collecte sont exposés). La réduction d'exposition, notée ***RE*** se calcule par : $\boldsymbol{RE}(T_B)=1-\mathbf{EX_{card}}(T_B)/|B|$

Les jeux de données sur les données personnelles sont intrinsèquement difficiles à obtenir. Nous présentons ainsi dans cette étude des résultats basés sur l'utilisation d'un générateur que nous avons développé, et nous avons utilisé des paramètres issus de véritables arbres de décision du domaine bancaire (Baesens *et al.*, 2003), et du domaine social, par le biais d'une collaboration avec le Conseil Général des Yvelines. Nous expliquons dans (Anciaux *at al.*, 2012) comment fonctionne ce générateur, basé sur les paramètres suivants : *D* l'ensemble des assertions, *R* l'ensemble des règles de collecte, *Q* l'ensemble des règles atomiques. Nous utilisons

[18] Voir : http://project.inria.fr/minExp/

une modélisation sous forme de graphe bipartite, et discutons dans la section suivante ses caractéristiques.

5.2. Mesures

Nous exécutons trois ensembles d'expériences. Nous choisissons une topologie du problème identique à celle que l'on obtient dans le cas réel du GEVA[19] : nous fixons d_R=4 (une règle de collecte est composée en moyenne de 4 règles atomiques) et d_D=4 (chaque assertion est présente en moyenne dans 4 règles atomiques). Puis nous faisons varier |*D*| et/ou |*R*|. Chaque point est la moyenne de plusieurs mesures pour réduire le biais statistique (i.e. chaque mesure est répétée 100 fois, sauf celles plus longues qu'une minute qui ont été exécutées 10 fois). Les expériences sont les suivantes :

Expérience 1: augmentation des documents. Nous faisons varier le nombre d'assertions nécessaires pour prouver les règles atomiques et nous fixons tous les autres paramètres. Nous considérons 10 règles de collecte. Concernant les paramètres du générateur, nous faisons varier |*D*| et fixons |*R*|=*10*, d_R=*4* et d_D=*4*. Nous donnons à *COUENNE* un temps d'exécution maximal de 2h. Nous donnons à *RAND** et *SA** le temps processeur équivalent à *une* exécution de *HME* sur la même instance. Notons que le temps d'exécution des algorithmes d'approximation est toujours inférieur à 10 minutes dans ce cas. Les résultats sont présentés en Figure 2 (pour la réduction d'exposition) et Figure 3 (pour le temps d'exécution).

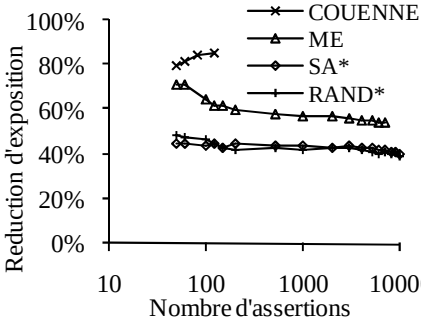

Figure 2. RE en variant le nombre d'assertions

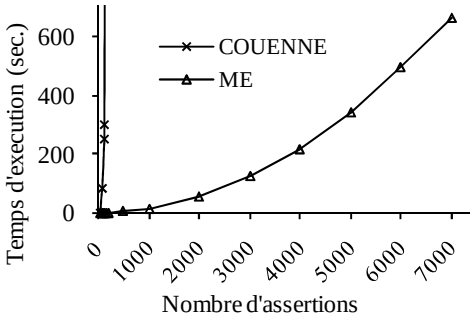

Figure 3. Temps d'exécution en variant le nombre d'assertions

Expérience 2: augmentation du nombre de règles de collecte. Nous faisons varier le nombre de règles de collecte et nous considérons un nombre fixe de *1000* assertions. Les paramètres sont donc : |*D*|=1000, d_R=4 et d_D=4. Dans ce cas, pour pouvoir faire varier |*R*| sur une large plage, il faut choisir une valeur de |*D*| suffisamment grande. De nouveau, on donne à *RAND** et *SA** le temps processeur

[19] Voir : http://www.cnsa.fr/IMG/pdf/GEVA_graphique-080529-2.pdf

équivalent à une exécution de *HME* (toujours moins de 10 minutes). *COUENNE* a été incapable de produire le moindre résultat en moins de 2h. Les résultats sont montrés à la Figure 4.

Expérience 3: augmentation simultanée du nombre d'assertions et de règles de collecte. Nous faisons varier le nombre de règles de collecte et d'assertions, tout en gardant un rapport constant $|D|/|R|=4$. Nous fixons $d_R=4$ et $d_D=4$. Cette expérience montre le comportement de l'algorithme lorsque la taille du jeu de données augmente, avec une topologie du problème donnée. La Figure 5 montre les résultats.

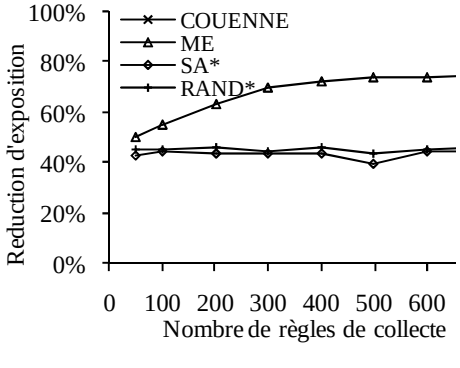

Figure 4. RE selon le nombre de règles de collecte

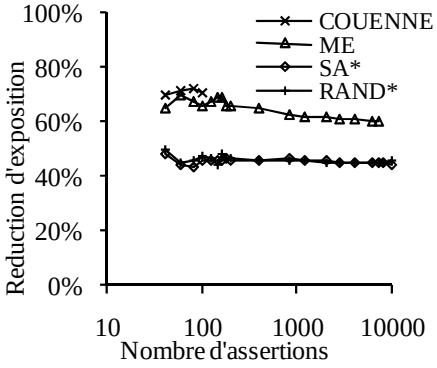

Figure 5. RE lorsque le nombre d'assertions et de règles de collecte augmente à rapport constant (=4)

Nous tirons les conclusions suivantes de cette expérimentation :

La réduction d'exposition est (presque toujours) conséquente: Selon la topologie du problème, les algorithmes apportent une réduction d'exposition de 30% à 80% par rapport à une implémentation classique de collecte limitée de données.

Une résolution exacte n'est pas souvent possible. Avec le temps limite de 2h, *COUENNE* n'arrive pas à trouver de résultat sur des problèmes de taille $|D| > 120$, dans l'expérience 1, 1 règle de collecte dans l'expérience 2, et 15 règles de collecte dans l'expérience 3. L'utilisation d'algorithmes d'approximation est incontournable pour obtenir des résultat pour un spectre très large de situations.

HME est le meilleur algorithme d'approximation. La réduction d'exposition obtenue avec *HME* est plus grande que celle obtenue par *RAND** et *SA**. Les résultats des Figures 2, 4 et 5 montrent que l'algorithme *HME* est meilleur que ses concurrents sur un large gamme de paramètres, avec une réduction d''environ 10%.

Pour conclure, l'implémentation de la collecte limitée de données en utilisant l'*Exposition Minimum* confère d'importants gains en terme de réduction d'exposition, et peut passer à l'échelle par l'utilisation d'algorithmes d'approximation qui fournissent de bons résultats.

6. Etat de l'art

La transposition de principes légaux dans des systèmes informatiques a été la base de nombreux travaux au cours de la dernière décennie. Des exemples emblématiques incluent la plateforme P3P (Cranor *et al.,* 2002) des langages de définition de politiques de vie privée comme EPAL (Ashley *et al.*, 2003) ou encore les bases de données Hippocratiques (Agrawal *et al.,* 2002). Le P3P permet de mettre en évidence des problèmes d'incompatibilité de politiques, mais ne permet pas de calibrer les données exposées, et ainsi d'atteindre la collecte limitée des données. D'autres langages de définition de politiques ont été proposés comme EPAL, XACML (Moses, 2005) ou WSPL (Anderson, 2004), mais à notre connaissance, aucun n'a été introduit avec la collecte limitée des données comme objectif. En revanche, l'architecture d'une base de données Hippocratique repose sur dix principes, qui incluent la collecte limitée des données. Elle adresse la collecte limité en maintenant un ensemble d'attributs qui sont requis pour atteindre un objectif donné. Toutefois, cette solution fait l'hypothèse que les données utiles et inutiles pour atteindre un objectif peuvent être déterminées au moment de la collecte. Comme nous l'avons montré dans l'introduction, si c'est peut être vrai pour certain cas simples, ce n'est jamais le cas en général pour des processus de décision complexes.

Un domaine produisant des résultats proches de ceux de notre étude est le domaine de la négociation automatique de confiance et le contrôle d'accès basé sur les autorisations, où les décisions d'accès se font sur une confrontation entre une politique de contrôle d'accès et un ensemble d'autorisations. Un petit nombre de travaux peuvent être considérés comme suivant l'approche de l'exposition minimale (Ardagna *et al.*, 2012 ; Chen *et al.*, 2005 ; Yao *et al.*, 2008). Tous ces travaux minimisent en effet l'exposition de données personnelles (sous la forme d'autorisations) tout en permettant une prise de décision (donner ou pas l'accès à une ressource). Toutefois le problème et les solutions sont différentes pour deux raisons essentielles. Tout d'abord, les processus de prise de décision que nous considérons sont bien plus complexes que le contrôle d'accès. Les règles de collecte que nous considérons modélisent des ensembles d'arbres de décision (classifieurs multi-classes) : de nombreuses dimensions peuvent être considérées (e.g. taux plus faible, durée plus longue, assurance réduite, etc.) dont chacune peut impacter l'offre finale proposée à l'utilisateur. D'autre part, dans notre contexte la prise de décision nécessite de très grandes quantités de données personnelles (e.g. un postulant à certaines allocations doit remplir des formulaires contenant plusieurs centaines de champs), tandis que dans le domaine du contrôle d'accès, seules quelques autorisations son prises en compte (e.g. jusqu'à 35 dans Ardagna *et al.,* 2012). Ainsi les résultats de ces travaux ne peuvent pas être réutilisés ici, puisqu'ils ne sont pas pertinents en terme d'expressivité et de passage à l'échelle.

7. Conclusion

Dans cet article, nous avons défini le concept d'*Exposition Minimum* des données personnelles, et étudié ses fondements théoriques, en l'exprimant sous la forme d'un problème SAT complexe. Nous avons étudié le spectre d'applicabilité des solutions générales de la recherche opérationnelle, en utilisant un solver de l'état de l'art. Dans les cas où une résolution exacte n'était pas possible, nous avons proposé plusieurs algorithmes approchés permettant de fournir un résultat approché, et expérimentalement de bonne qualité, en temps polynomial. Dans tous les cas, nous avons montré qu'une réduction d'exposition de l'ordre de 50% par rapport à une stratégie classique pouvait être obtenue. Cette réduction sert à la fois les intérêts de l'utilisateur, qui voit ses données personnelles moins divulguées, et le fournisseur de service, qui voit ses coûts de traitement réduits, et ses risques amoindris dans le cas d'une fuite de données.

Le principe de l'Exposition Minimum présenté dans cet article sera utilisé dans le cadre de la dématérialisation de certaines demandes d'aides sociales par le Conseil Général des Yvelines. Nous avons en effet présenté dans (Anciaux *et al.*, 2013) un prototype exécutant une minimisation de données pour un scénario d'aide sociale montrant la faisabilité de l'approche dans le cas où les règles de collecte sont privées également, et qui nécessite donc l'utilisation de matériel sécurisé.

Remerciements

Ces travaux ont été financés par les projets ANR DEMOTIS (ANR-08-SEGI-007), ANR KISS (ANR-11-INSE-0005), DIGITEO LeTeVoNe et INRIA Project Lab CAPPRIS. Nous remercions Jean-François Navarre du Conseil Général des Yvelines pour les discussions autour des scénarios d'aide sociale.

Bibliographie et références

Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. (2002). Hippocratic databases. In *Proceedings of the 28th International Conference on Very Large Data Bases*.

Alimonti, P., Ausiello, G., Giovaniello, L., and Protasi, M. (1998). *On the Complexity of approximating weighted satisfiability problems*. Technical Report, Università di Roma.

Allard, T., Anciaux, N., Bouganim, L., Guo, Y., Le Folgoc, L., Nguyen, B., Pucheral, P., Ray, I., Ray, I., and Yin, S. (2012). Secure Personal Data Servers: a Vision Paper. In *VLDB Endowment*, *3*(1).

Anciaux, N., Bezza, W., Nguyen, B., and Vazirgiannis, M. (2013) MinExp-Card: Limiting Data Collection Using a Smart Card. In *Proceedings of the 16th International Conference on Extending Database Technology*.

Anciaux, N., Nguyen, B., and Vazirgiannis, M. (2012). *Minimum Exposure in classification scenarios*. INRIA Research Report. Available at http://blog.inria.fr/minexp/

Anciaux, N., Nguyen, B., and Vazirgiannis, M. (2011) Limiting Data Collection in Application Forms. In *Proceedings of the 10th IEEE Annual Conference on Privacy Security and Trust*.

Anderson, A.H. (2004). An Introduction to the Web Services Policy Language (WSPL). In *Proceedings of the POLICY Workshop*.

Ardagna, C.A., De Capitani di Vimercati, S., Foresti, S., Paraboschi, S., and Samarati, P. (2012). Minimising Disclosure of Client Information in Credential-Based Interactions. *Int. Journal of Information Privacy, Security and Integrity*, *1*(2/3):205-233.

Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. (2003). *Enterprise privacy authorization language 1.2 (EPAL 1.2)*. W3C Member Submission.

Baesens, B., Setiono, R., Mues, C. and Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, *49*(3):312-329.

Belotti, P., Lee, J., Liberti, L., Margot, F., and Wachter, A. (2009). Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software, 24*(4-5):597-634.

Chen, W., Clarke, L., Kurose, J., and Towsley, D. (2005). Optimizing cost-sensitive trust-negotiation protocols. *IEEE Computer and Communications Societies*.

Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J. (2002). *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. W3C Recommendation.

Crook, J.N., Edelman, D.B., and Thomas, L.C. (2007). Recent developments in consumer credit risk assessment. *Eur*o. *J. of Op. Research*, *183*(3):1447-1465.

Escoffier, B., and Paschos, V. Th. (2007). Differential approximation of MIN SAT. *European Journal of Operational research*, 181(2):620-633.

European Parliament. (1995). Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data. *Official Journal of the EC, 23*.

Fourer, R., Gay, D.M., and Kernighan, B.W. (1990). A Modeling Language for Mathematical Programming. *Management Science, 36*.

Huysmans, J., Baesens, B., and Vanthienen, J. (2007). *Using rule extraction to improve the comprehensibility of predictive models*. Open Access publications from Katholieke Universiteit Leuven.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

Moses, T. (2005). *Extensible access control markup language (XACML) version 2.0*. Oasis Standard.

OECD (1980). Guidelines on the Protection of Privacy and Transborder Flows of Personal Data.

Papadimitriou, C., and Yannakakis, M. (1988) Optimization, approximation and complexity classes. In *Proceedings of the 20th ACM Symposium on the Theory of Computer Science*.

Ponemon Institute, LLC. (2011). 2010 Annual Study: U.S. Cost of a Data Breach.

Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering, 13*(6):1010-1027

Sweeney, L. (2002). k-Anonymity: a model for protecting privacy. *Int. Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5)*:557-570.

Xiao, X., and Tao, Y. (2006). Personalized privacy preservation. In *Proceedings of ACM Special Interest Group on the Management Of Data*.

Yao, D., Frikken, K.B., Atallah, M.J., and Tamassia, R. (2008). Private information: To reveal or not to reveal. In *ACM Transactions on Information and System Security, 12*(1), article 6.