par N. Anciaux et B. Nguyen

Limiter la collecte des données personnelles
Un problème juridique NP-difficile

Oui, un problème mathématique tout à fait théorique à l'origine peut avoir une application très pratique ! La preuve avec le problème de *n*-exposition, qui concerne la collecte de données personnelles, et la problématique de l'attribution d'aides sociales. Un défi avec données réelles est proposé aux lecteurs.

Le concept de protection des données personnelles existe depuis longtemps. En France, par exemple, la loi relative à l'informatique, aux fichiers et aux libertés, dite *loi informatique et libertés*, date de 1978 et pose l'existence d'un certain nombre de principes fondamentaux liés au traitement des données personnelles, protégeant ainsi la vie privée des citoyens. Parmi ses nombreux principes, l'un d'entre eux régit le *traitement* des données personnelles : la collecte limitée de données personnelles en vue d'un traitement informatique (voir en encadré).

Traitement des données personnelles

Un *traitement de données personnelles* est un terme général, correspondant à un traitement automatique (informatique) ou non portant sur des données personnelles, quel que soit le procédé utilisé, et notamment la collecte, l'enregistrement, l'organisation, la conservation, l'adaptation ou la modification, l'extraction, la consultation, l'utilisation, la communication par transmission, diffusion ou toute autre forme de mise à disposition, le rapprochement ou l'interconnexion, ainsi que le verrouillage, l'effacement ou la destruction.

En effet, la loi de 1978 énonce dans ses articles 6 (sur la collecte et conservation des données) et 7 (sur le consentement) les conditions de licéité d'un traitement (informatique). On voit que la *finalité* du traitement doit être *« déterminée, explicite et légitime »*, que les données traitées doivent être *« adéquates, pertinentes et non excessives »*, mais également *« exactes et complètes »*. En d'autres termes, il n'est pas légal de collecter des données sans savoir si elles seront utiles ou pas pour le processus.

Or, nous allons voir que ce principe (juridique) de *minimalité* des données traitées est très difficile à mettre en place de manière pratique puisqu'il pose des problèmes mathématiques (informatiques) complexes : il est en effet NP-difficile de décider, compte tenu d'un traitement modélisé sous la forme de règles logiques permettant une prise de décision multicritère, si un ensemble de données collecté est minimal ou non.
À un problème de décision NP-complet (voir l'article consacré à ce thème dans ce hors-série), on peut lier un problème d'optimisation qui cherche à calculer la

plus petite valeur acceptable pour n ; ce nouveau problème est qualifié dans ce cas de NP-difficile, et sa résolution est au moins aussi coûteuse que la résolution du problème de décision NP-complet associé.

Pour illustrer le problème, prenons comme exemple de traitement automatique de données personnelles un système informatique permettant d'attribuer des prêts bancaires personnalisés. Une banque propose des prêts à la consommation de 5 000 euros à 10 % sur trois ans, personnalisables selon quatre critères : montant plus élevé du prêt, réduction du taux, augmentation de la durée de remboursement et réduction de l'assurance liée au prêt. La décision d'attribuer chacun de ces quatre critères est prise sur la base d'un ensemble de règles logiques, dit *modèle métier* de la banque, exprimées avec des variables Booléennes $b_{k,i,j}$, où chaque règle r_k permettant d'obtenir un bénéfice c_k est en forme normale disjonctive, et où certaines variables booléennes $b_{k,i,j}$ interviennent dans plusieurs règles. Dans l'exemple de l'encadré ci-contre, on a $b_{1,1,1} = b_{3,1,1} = b_{4,2,1} = p_1$, $b_{1,1,2} = b_{3,2,1} = b_{4,1,1} = p_2$ et ainsi de suite. Il est intéressant de noter que, dans de nombreux cas, les règles sont issues de techniques de fouille de données (*data mining*) sur la base de décisions prises précédemment par des experts.

Considérons la règle r_1 :
$(p_1 \wedge p_2) \vee (p_3 \wedge p_4) \Rightarrow c_1$.
Elle exprime le fait qu'un client peut obtenir un prêt plus élevé (l'avantage c_1) si son revenu annuel est supérieur à 30 000 $ et si son patrimoine est supérieur à 100 000 $ ou bien s'il possède une assurance vie et un nantissement supérieur à 100 000 $. Afin de bénéficier de cet avantage, un client doit exposer un ensemble d'assertions logiques $as_1, as_2, as_3 \ldots$ supposées de la forme *attribut* = *valeur*, tel que cet ensemble permet de prouver que le corps de la règle $(p_1 \wedge p_2) \vee (p_3 \wedge p_4)$ est vrai.
Dans l'exemple, $as_1 \Rightarrow p_1$, $as_2 \Rightarrow p_2$, $as_3 \Rightarrow p_3$ et $as_4 \Rightarrow p_4$. Il est donc évident que le corps de règle r_1 est vrai, et donc que ce client peut bénéficier d'un montant de prêt plus élevé. Toutefois, il est inutile que le client transmette l'intégralité de ces informations pour bénéficier de cet avantage (on dit également qu'il *expose* ces informations). Il est en réalité nécessaire et suffisant de ne transmettre que $\{as_1, as_2\}$ ou $\{as_3, as_4\}$ afin de limiter la collecte de données personnelles (pour simplifier, on considère que chaque attribut a la même sensibilité qu'un autre ; il est ainsi aussi « problématique » pour le client de transmettre son âge que son adresse). On préfère donc transmettre le plus petit nombre d'attributs.

Un exemple de prêt bancaire

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.

Assertions Client :

as_1 : revenu_annuel = $35.000, as_2 : patrimoine = $150.000,
as_3 : nantissement = $175.000, 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.

L'article 6 de la loi informatique et libertés

« Un traitement ne peut porter que sur des données à caractère personnel qui satisfont aux conditions suivantes :

1) Les données sont collectées et traitées de manière loyale et licite ;
2) Elles sont collectées pour des finalités déterminées, explicites et légitimes et ne sont pas traitées ultérieurement de manière incompatible avec ces finalités. Toutefois, un traitement ultérieur de données à des fins statistiques ou à des fins de recherche scientifique ou historique est considéré comme compatible avec les finalités initiales de la collecte des données, s'il est réalisé dans le respect des principes et des procédures prévus au présent chapitre, au chapitre IV et à la section 1 du chapitre V ainsi qu'aux chapitres IX et X et s'il n'est pas utilisé pour prendre des décisions à l'égard des personnes concernées ;
3) Elles sont adéquates, pertinentes et non excessives au regard des finalités pour lesquelles elles sont collectées et de leurs traitements ultérieurs ;
4) Elles sont exactes, complètes et, si nécessaire, mises à jour ; les mesures appropriées doivent être prises pour que les données inexactes ou incomplètes au regard des finalités pour lesquelles elles sont collectées ou traitées soient effacées ou rectifiées ;
5) Elles sont conservées sous une forme permettant l'identification des personnes concernées pendant une durée qui n'excède pas la durée nécessaire aux finalités pour lesquelles elles sont collectées et traitées.

S'il n'y a qu'une seule règle de collecte, le problème est simple : il suffit de transmettre la conjonction impliquant le moins d'attributs. Le problème devient plus difficile lorsque le client souhaite bénéficier de *plusieurs* avantages c_k simultanément.

Afin de définir formellement le problème mathématique de la minimisation des données collectées (dit également *problème de n-exposition*), posons $E_R = \wedge_k(r_k) = \wedge_k(\vee_i(\wedge_j b_{k,i,j}))$, appelée *formule booléenne de l'ensemble de règles*. Le problème s'énonce formellement de la manière suivante :

On se donne un ensemble de règles $R = \{r_1, r_2, r_3 \ldots\}$, un ensemble d'assertions $\text{data}_u = \{as_1, as_2, as_3 \ldots\}$ tel que tous les r_k sont vrais, un ensemble de variables booléennes $B = \{p_1, p_2 \ldots p_q\}$ tel que p_m = vrai $\Leftrightarrow$ as_x est transmis, et enfin la formule booléenne de l'ensemble de règles R, notée $E_R = \wedge_k(\vee_i(\wedge_j b_{k,i,j}))$ où, quels que soient les indices k, i et j, $b_{k,i,j} \in B$. On dit que data_u est *n-exposable par rapport à* R si, et seulement si, il existe une affectation de valeurs booléennes aux variables de B telle que E_R est vraie, et le nombre de $p_1, p_2 \ldots p_m$ prenant la valeur vraie est inférieur n. Décider si l'ensemble data_u est n-exposable par rapport à R.

Pour les données présentées dans l'encadré précédent, on peut facilement vérifier que la solution
$T_B = \{p_1 = V, p_2 = V, p_3 = F, p_4 = F, p_5 = V, p_6 = V, p_7 = V, p_8 = F, p_9 = F, p_{10} = F\}$ valide bien E_R, et donc que R est 5-exposable. Pour montrer que 5 est bien la plus petite valeur acceptable, une manière est de tester toutes les affectations ayant quatre valeurs vraies (il en existe C_{10}^4 soit 210), et montrer qu'aucune ne convient.

Il est possible de montrer que ce problème, dans sa généralité, est NP-complet, en se fondant sur une réduction à un problème assez connu en logique : le *problème de satisfiabilité minimale avec pondération* (ou Min Weighted Sat). On définit le problème d'optimisation associé, qui cherche à trouver le plus petit entier m pour lequel l'ensemble des p_m doivent être vrais pour que E_R soit vraie. Le problème d'optimisation est NP-difficile, ce qui signifie qu'il peut être très coûteux de trouver le résultat exact à cette question pour de grandes valeurs de m. Pour calculer la solution exacte du problème, on peut utiliser une méthode

de force brute, c'est-à-dire tester toutes les solutions possibles : il y en a très exactement 2^q où q représente le nombre de prédicats booléens du problème. Le coût de cette méthode est donc hautement prohibitif.
Une autre manière de faire est d'utiliser un outil de résolution exact de problèmes d'optimisation. Le problème doit alors être écrit dans un langage particulier : le langage AMPL (langage de modélisation pour la programmation mathématique), qui permet d'exprimer une fonction à minimiser par rapport à un ensemble de variables (ici les prédicats booléens) et de contraintes (ici les règles). Un exemple d'un tel programme est donné en encadré. Les contraintes contiennent des multiplications, ce qui rend le problème non-linéaire. Tant que le nombre de variables n'est pas trop important (disons une centaine), ce programme peut être résolu par des logiciels comme Couenne (pour Convex over and under envelopes for non-linear estimation, disponible en ligne et *open-source*).

Une dernière approche est de proposer des algorithmes de complexité polynomiale, et donc faciles et rapides à calculer, permettant de trouver un résultat approché de la solution. L'encadré qui suit donne l'exemple d'un algorithme aléatoire, nommé RAND*, très simple, qui permet de calculer une solution acceptable au problème, en générant aléatoirement N solutions acceptables, puis en choisissant la meilleure. Toutefois, la minimalité de la solution n'est aucunement garantie !
Le lecteur est invité à chercher d'autres algorithmes pour résoudre le problème de manière approchée, et à comparer, à temps de calcul égal, leur qualité avec l'algorithme naïf RAND*. Ainsi, on voit que le problème de limiter la collecte de données est compliqué, puisque la simple identification d'une donnée potentiellement utile est un problème informatique difficile et coûteux en temps !

N. A. & B. N.

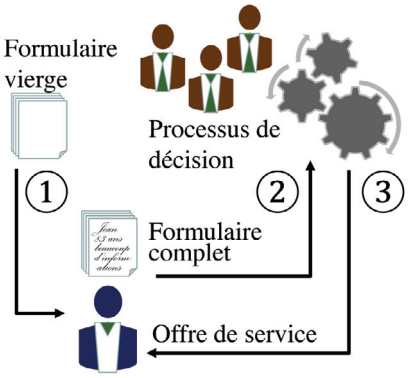
Un exemple de programme en AMPL

```
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;
```

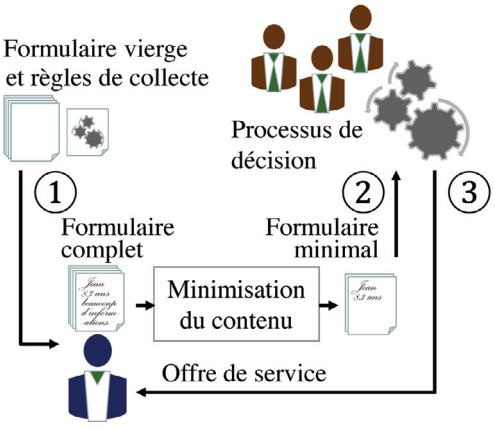
L'algorithme approché RAND*

Entrées : ensemble de K règles r_k contenant q prédicats distincts p_i
nombre de passes N
Sortie : meilleur ensemble de valeurs de vérité p_i trouvé
Variables : $p\_optimal$, un tableau de q booléens
x, une variable de type entier

1. Mettre à VRAI toutes les valeurs de $p\_optimal_i$
2. Pour *compteur* allant de 1 à N faire
3. Mettre toutes les valeurs de p_i à FAUX
4. Pour k allant de 1 à K faire
5. $x \leftarrow$ une valeur aléatoire comprise entre 1 et le nombre de disjonctions dans r_k
6. Pour j allant de 1 au nombre de prédicats dans la $x^{ème}$ disjonction de r_k faire
7. mettre à VRAI la valeur du prédicat p_i correspondant à $b_{k,x,j}$
8. FinPour
9. FinPour
10. Si le nombre de prédicats p_i mis à VRAI est plus petit que le nombre de prédicats $p\_optimal_i$ mis à VRAI, alors
11. Pour i allant de 1 à q faire
12. $p\_optimal_i \leftarrow p_i$
13. FinPour
14. FinSi
15. FinPour
16. Retourner la liste des $p\_optimal_i$



Processus classique
Le demandeur renseigne l'intégralité du formulaire d'application, transmettant ainsi toutes les informations identifiées comme ayant un impact potentiel sur le processus de prise de décision.



Processus à exposition minimale.
Les informations renseignées par le demandeur sont minimisées conformément aux règles de collecte, ce qui conduit à prendre la même décision, à partir d'un formulaire contenant (beaucoup) moins d'informations.

Processus de prise de décision à exposition minimale d'information.

Application : l'attribution de l'aide sociale

En France, les Conseils généraux sont responsables d'attribuer une aide sociale aux personnes dépendantes, sous forme de support financier, matériel ou humain. L'aide est sollicitée au travers d'un ensemble de formulaires à renseigner décrivant la situation précise de la personne dépendante, et comportant plusieurs centaines de champs à saisir avec l'aide de son médecin, de son assistant social et de son entourage. Ainsi, les Conseils généraux traitent chacun plusieurs dizaines de milliers de demandes par an, nécessitant l'intervention de centaines d'employés dépouillant les formulaires pour extraire et vérifier les informations, et décider de la forme que devra prendre l'aide afin de répondre au mieux à la situation du requérant. Un processus identifiant, en amont du traitement de la demande, l'ensemble minimal d'assertions permettant de conduire à la décision limite la quantité de données personnelles exposées par le requérant conformément à la loi, et accélère le traitement des demandes.

Le défi. Ainsi, dans le cas du Conseil général des Yvelines, le jeu de règles obtenu implique 440 prédicats et conduit à 63 bénéfices potentiels. Ces règles ont été transposées au format AMPL (données disponibles sur le site https://project.inria.fr/minexp/, rubrique Challenge). Le solveur Couenne lancé sur la modélisation non linéaire ne parvient pas à trouver une solution minimale, même après plusieurs jours de fonctionnement. L'algorithme RAND* permet d'obtenir une solution en quelques secondes (230 éléments sont conservés sur 440, avec N = 1 000, et 215 éléments avec N=100 000). Cette solution se situe probablement assez loin de la solution optimale. Nos lecteurs sont invités à développer et comparer entre eux d'autres algorithmes basés sur des heuristiques plus efficaces, permettant de parvenir à une solution plus proche de l'optimale, tout en restant très rapides !

Références

- *Exposition minimum de données pour des applications à base de classifieurs*. Nicolas Anciaux, Benjamin Nguyen et Michalis Vazirgiannis, *Ingénierie des Systèmes d'Information* 18(4), 2013.
- Le défi : https://project.inria.fr/minexp/, rubrique Challenge (en anglais).