Vers l'exploitation de grandes masses de données

Raphaël Féraud, Marc Boullé, Fabrice Clérot , Françoise Fessant

France Télécom R&D, avenue Pierre Marzin, 22307 Lannion
Contact : raphael.feraud@orange-ftgroup.com

Résumé : Une tendance lourde depuis la fin du siècle dernier est l'augmentation exponentielle du volume des données stockées. Cette augmentation ne se traduit pas nécessairement par une information plus riche puisque la capacité à traiter ces données ne progresse pas aussi rapidement. Avec les technologies actuelles, un difficile compromis doit être trouvé entre le coût de mise en œuvre et la qualité de l'information produite. Nous proposons une approche industrielle permettant d'augmenter considérablement notre capacité à transformer des données en information grâce à l'automatisation des traitements et à la focalisation sur les seules données pertinentes.

Mots clés : fouille de données, grande volumétrie, sélection de variables, sélection d'instances.

1 Introduction

Selon Fayyad et al (1996), le Data Mining est un processus non-trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données. Plusieurs intervenants industriels ont proposé une formalisation de ce processus, sous la forme d'un guide méthodologique nommé CRISP-DM pour CRoss Industry Standard Process for Data Mining, voir Chapman et al (2000). Le modèle CRISP-DM (FIG 1) propose de découper tout processus Data Mining en six phases:

1. La phase de *recueil des besoins* fixe les objectifs industriels et les critères de succès, évalue les ressources, les contraintes et les hypothèses nécessaires à la réalisation des objectifs, traduit les objectifs et critères industriels en objectifs et critères techniques, et décrit un plan de résolution afin d'atteindre les objectifs techniques.
2. La phase de c*ompréhension des données* réalise la collecte initiale des données, en produit une description, étudie éventuellement quelques hypothèses à l'aide de visualisations et vérifie le niveau de qualité des données.
3. La phase de *préparation des données* consiste en la construction d'une table de données pour modélisation (Pyle, 1999; Chapman et al, 2000). Nous nous y intéressons plus particulièrement par la suite.
4. La phase de *modélisation* procède à la sélection de techniques de modélisation, met en place un protocole de test de la qualité des modèles obtenus, construit les modèles et les évalue selon le protocole de test.
5. La phase *d'évaluation* estime si les objectifs industriels ont été atteints, s'assure que le processus a bien suivi le déroulement escompté et détermine la phase suivante.

6. La phase de *déploiement* industrialise l'utilisation du modèle en situation opérationnelle, définit un plan de contrôle et de maintenance, produit un rapport final et effectue une revue de projet.

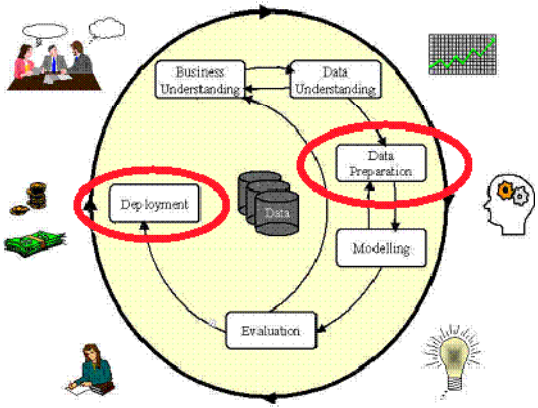

FIG 1: Processus Data Mining CRISP DM

Le modèle CRISP-DM est essentiellement un guide méthodologique pour la conduite d'un projet Data Mining. La plupart des praticiens du Data Mining s'accordent pour dire que les phases de préparation de données et de déploiement consomment à elles seules 80 % des ressources des projets. L'explication est simple. L'utilisation des méthodes statistiques nécessite de représenter les données sous la forme d'un tableau croisé : en ligne les instances et en colonnes les variables caractérisant ces instances. Or, afin d'optimiser le stockage, les données sont stockées dans des bases de données relationnelles, et ce quel que soit le phénomène étudié : les gènes, les transactions de cartes bancaires, les sessions IP, les informations sur les clients…

Lors de la phase de *préparation de données*, la première tâche de l'analyste est donc d'extraire un tableau croisé du système d'information. Cette étape n'est pas anodine car le nombre de représentations potentielles des données relationnelles dans un tableau croisé est gigantesque (FIG 2). En pratique, l'analyste doit faire un choix a priori de l'ensemble des variables explicatives sur lesquelles se fera l'étude. La conséquence est que la perte d'information due à la mise à plat des données relationnelles est très importante.

Lors de la phase de *déploiement,* le modèle construit préalablement doit être appliqué à toute la population concernée, afin de produire un score pour chaque instance. Toutes les variables explicatives pour toutes les instances doivent être construites. Cette étape est potentiellement très couteuse lorsque le nombre d'instances et de variables explicatives est important.

Les principaux produits commerciaux de Data Mining, comme SAS, ou SPSS, proposent des plateformes permettant de construire et de déployer des modèles prédictifs. Néanmoins, ils n'offrent pas de solution satisfaisante pour exploiter tout le potentiel de l'information contenue dans la base de données source. Dans les applications industrielles construites sur ces plateformes logicielles, le nombre de variables explicatives à partir desquelles sont construits les modèles reste limité à quelques centaines. Or le potentiel est tout simplement d'un autre ordre. Dans l'exemple illustratif présenté (FIG 2), la base de données ne contient

que deux tables. L'étude du nombre d'usages par type de service, par mois, et par jour de la semaine pourrait conduire à elle seule à construire 10000 variables explicatives !

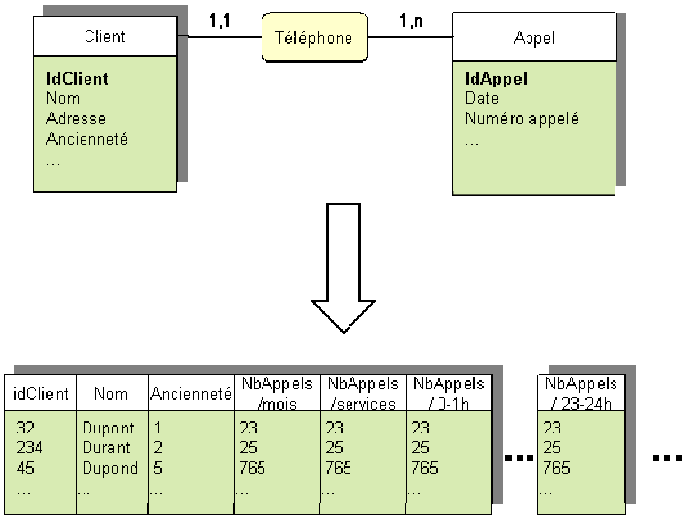

FIG 2: La mise à plat de données relationnelles impose de choisir a priori un sous-ensemble d'agrégats, ce qui conduit à une perte d'information.

Avec ces technologies, un difficile compromis doit être trouvé entre le coût de mise en œuvre et la qualité de l'information produite.

Parmi les produits commerciaux, KXEN propose un module permettant de construire automatiquement des agrégats à partir de données temporelles. L'avantage est de pouvoir explorer un plus grand nombre de variables explicatives. Nous avons très largement généralisé et systématisé cette approche pour automatiser entièrement le processus de préparation de données. La construction de variables explicatives est entièrement pilotée par un algorithme de sélection de représentation très performant (Boullé, 2007). Dans la section suivante nous décrirons les principaux éléments de cette architecture de traitement de données[1] avant de détailler l'algorithme de sélection de représentation.

Pour faciliter le déploiement des modèles, nous proposons une méthode permettant d'extraire d'une base complète, une table réduite de parangons[2]. Cette table de parangons est constituée des seules variables explicatives pertinentes au sens du score construit et des instances les plus représentatives des variables sélectionnées. La table de parangons est reliée à la base complète par un index construit automatiquement. Toute l'information produite sur la table des parangons peut être déployée par une simple jointure sur l'ensemble des instances. Les algorithmes permettant d'extraire les parangons et de les indexer efficacement sont décrits dans la section 3.

Avant de conclure, nous présentons dans la section 4 des résultats expérimentaux très encourageants obtenus sur des données réelles du groupe France Télécom.

[1] Système de pilotage d'extraction de tableaux croisés, dépôt le 01/09/2006, numéro INPI 06 07965

[2] Procédé et dispositif de construction et d'utilisation d'une table de profils réduits de parangons, dépôt le 27 mai 2005, numéro INPI 05 05412

2 L'automatisation de la préparation de données

2.1 Généralités

L'objectif de la sélection de représentation est triple: améliorer la performance prédictive des modèles, le temps d'apprentissage et de déploiement des modèles, et permettre leur interprétation (Guyon et Elisseeff, 2003). La sélection de variables est un sujet bien couvert dans la recherche en fouille de données, si bien qu'aujourd'hui, les méthodes de sélection de variables sont suffisamment robustes pour permettre la construction de modèle en très grande dimension (Guyon et al, 2006), y compris lorsque le nombre de variables est grand devant le nombre d'instances. Afin de porter ces résultats sur des applications industrielles, nous proposons une architecture de traitement permettant à un algorithme de sélection de représentation de piloter des extractions de tables de données de grande taille. Nous présenterons ensuite, la méthode MODL (Boulle, 2007), à la fois robuste et rapide, utilisée pour sélectionner la meilleure représentation.

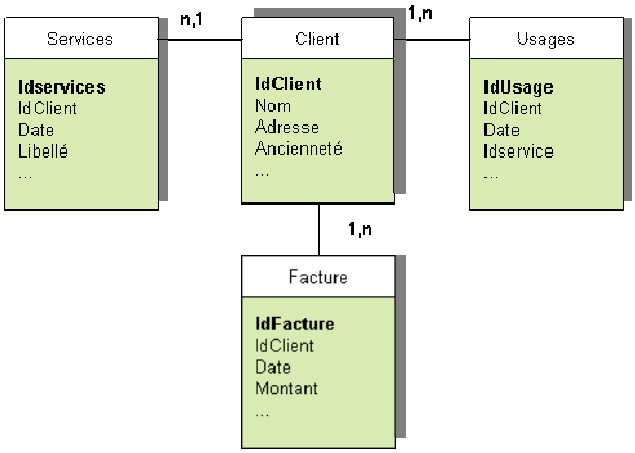

FIG 3 : La table principale est la table client. Les clients possèdent des services, utilisent ces services et paient des factures.

2.2 L'architecture de traitements

Contrairement à l'architecture actuelle de la fouille de données, les variables explicatives ne sont pas calculées à l'avance dans un datamart. Dans notre architecture de traitements, les données permettant de construire les variables explicatives sont stockées dans une base de données relationnelle simple, le data folder (FIG 3). Les variables explicatives sont construites et sélectionnées automatiquement en fonction de l'étude menée. Le modèle de données du data folder permet d'assurer une normalisation des différentes sources de données qui seront toujours présentées sous la forme d'un schéma en étoile :

- La table principale correspond au domaine étudié. Pour l'analyse de données clients, cette table comprendra les informations directement liées au client comme son nom, son adresse.

- Les tables de faits sont en liens multiples avec la table principale. A chaque instance de la table principale correspond un nombre variable de faits. Pour des données de télécommunications on retrouvera par exemple une table décrivant les services détenus, une table traçant les usages de ces services, une table récapitulant les factures.

Ce type de modélisation est suffisamment expressif pour s'adapter à tous les types de données. Sa structure en étoile permet d'optimiser le calcul des variables lorsque la clé d'agrégation est sur la table principale. Dans ce cas, le calcul d'une variable nécessite au plus une seule jointure entre la table principale et la table de faits alors que dans un entrepôt de données, le calcul d'une variable peut impliquer un très grand nombre de jointures. Enfin ce type de modèle de données, où le rôle de chaque table est bien défini, permet de construire des langages d'interrogation formatés et donc automatisables.

2.3 Le pilotage des extractions

Les extractions entre le data folder et les tableaux croisés sont paramétrées par trois types de dictionnaires :

- le dictionnaire de sélection pour filtrer les instances,
- le dictionnaire de requêtes pour spécifier les mises à plat des données du data folder,
- le dictionnaire de préparation pour spécifier le recodage des variables.

Ces dictionnaires d'extraction permettent de définir des requêtes suffisamment simples pour être pilotées automatiquement par les processus de sélection de représentation et suffisamment expressives pour produire une très grande variété de variables explicatives.

Quelque soit son objet, une requête portera toujours sur deux tables au plus : la table principale et une des tables de faits. La clé d'agrégation se trouvera toujours sur la table principale. Les opérateurs d'agrégation et de sélection porteront sur les champs des tables de faits ou sur les champs de la table principale.

Par exemple pour produire le nombre d'utilisations de chaque service par jour nommé pour tous les clients, il suffira de spécifier que la requête porte sur la table Usages, qu'elle fait appel à l'opérateur de comptage et qu'elle consiste à croiser les champs JourNommé(Date) et Libellé(IdService). En une seule ligne, il est ainsi possible de spécifier plusieurs milliers de variables explicatives.

2.4 La sélection de représentation

L'architecture de traitements permet à un algorithme de piloter efficacement des extractions de tableaux croisés pouvant compter des dizaines de milliers de variables. Pour sélectionner la meilleure représentation possible, nous avons besoin d'une méthode de sélection de variable particulièrement robuste et rapide. Deux approches principales, filtre et enveloppe (Kohavi et John, 1997), ont été proposées dans la littérature pour sélectionner les variables. Les méthodes enveloppes sont très coûteuses en temps de calcul (Féraud et Clérot, 2001, Lemaire et Féraud, 2006). C'est pourquoi nous avons retenu une approche de type filtre pour déterminer et construire les variables explicatives pertinentes. L'approche filtre la plus fréquemment utilisée repose sur la mise en œuvre de tests statistiques (Saporta, 1990), comme par exemple le test du Khi2 pour les variables explicatives catégorielles, ou les tests

de Student ou de Fisher-Snedecor pour les variables explicatives numériques. Ces tests d'indépendance sont simples à mettre en œuvre, mais présentent de nombreux inconvénients. Ils se limitent à une discrimination entre variables dépendantes et indépendantes, sans permettre un ordonnancement précis des variables explicatives, et sont contraints par des hypothèses d'applicabilité fortes (effectifs minimaux, hypothèse de distribution gaussienne dans le cas numérique…). De nombreux autres critères d'évaluation de la dépendance entre deux variables ont été étudiés dans le contexte des arbres de décision (Zighed et Rakotomalala, 2000). Ces critères sont basés sur une partition de la variable explicative, en intervalles dans le cas numérique et en groupe de valeurs dans le cas catégoriel. En recherchant de façon non paramétrique un modèle de dépendance entre variables explicatives et cible, ils permettent une évaluation fine de l'importance prédictive des variables explicatives. Dans le cas où tous les modèles de partitionnement de la variable explicative sont envisagés, un compromis doit être trouvé entre finesse de la partition et fiabilité statistique. Ce compromis est réalisé dans l'approche MODL (Minimum Optimized Description Length, voir Boullé, 2005, 2006) en formulant le problème comme un problème de sélection de modèles et en adoptant une approche bayesienne. Les résultats obtenus par l'approche MODL sont particulièrement convaincants sur le dernier Performance Prediction Challenge[3].

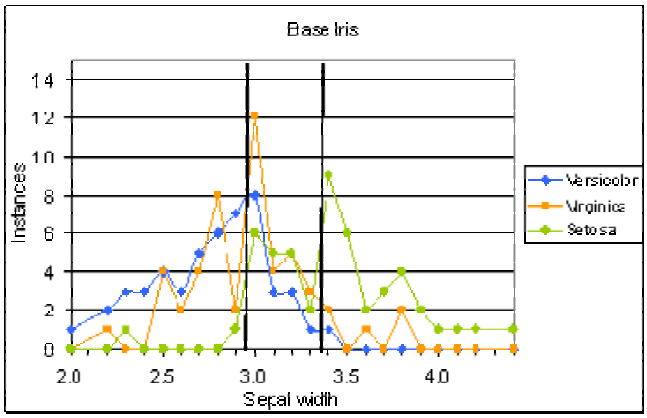

FIG 4 Discrétisation MODL de la variable largeur de sépale en trois intervalles pour la classification de la base Iris en trois classes.

Cette approche est valable aussi bien pour les variables numériques que pour les variables catégorielles. Les variables sont partionnées, en intervalles dans le cas numérique et en groupes de valeurs dans le cas catégoriel. Dans le cas numérique, il s'agit d'un modèle de discrétisation, qui correspond à une liste d'intervalles auxquels sont associées les fréquences de la variable cible. Les paramètres d'une discrétisation particulière sont le nombre d'intervalles, les bornes des intervalles et les effectifs des classes cibles par intervalle. Une approche bayesienne est appliquée pour construire un critère permettant de sélectionner le

[3] http://www.modelselect.inf.ethz.ch/index.php

meilleur modèle de discrétisation : le plus probable connaissant les données. Dans le cas d'une discrétisation, le critère à optimiser obtenu est :

$$\log(N) + \log\left(C_{N+I-1}^{I-1}\right) + \sum_{i=1}^{I} \log\left(C_{N_{i.}+J-1}^{J-1}\right) + \sum_{i=1}^{I} \log\left(N_{i.}!/N_{i1}!N_{i2}!...N_{iJ}!\right)$$

où N est le nombre d'individus, J le nombre de classes cibles, I le nombre d'intervalle, $N_{i.}$ le nombre d'individus dans l'intervalle i et N_{ij} le nombre d'individus de la classe j dans l'intervalle i.

Les trois premiers termes représentent l'a priori du modèle: choix du nombre d'intervalles, des bornes des intervalles, et de la distribution des valeurs cibles dans chaque intervalle. Le dernier terme représente la vraisemblance d'observer les valeurs de la variable cible connaissant le modèle de discrétisation. A titre illustratif (FIG 4), nous donnons le modèle de discrétisation de la variable SepalWidth obtenu par l'optimisation de ce critère pour séparer les différents iris (Blake et Merz, 1996).

Le même type de critère est construit pour le groupement de valeurs. Le critère possède une structure similaire à celle du critère de discrétisation, en remplaçant dans les deux premiers termes la probabilité a priori d'une partition en intervalles par celle d'une partition en groupes de valeurs.

$$\log(V) + \log\left(B(V,I)\right) + \sum_{i=1}^{I} \log\left(C_{N_{i.}+J-1}^{J-1}\right) + \sum_{i=1}^{I} \log\left(N_{i.}!/N_{i1}!N_{i2}!...N_{iJ}!\right)$$

où $B(V,I)$ est le nombre de répartitions des V valeurs explicatives en I groupes.

Valeur	EDIBLE	POISONOUS	Effectif
BROWN	55.2%	44.8%	1610
GRAY	61.2%	38.8%	1458
RED	40.2%	59.8%	1066
YELLOW	38.4%	61.6%	743
WHITE	69.9%	30.1%	711
BUFF	30.3%	69.7%	122
PINK	39.6%	60.4%	101
CINNAMON	71.0%	29.0%	31
GREEN	100.0%	0.0%	13
PURPLE	100.0%	0.0%	10

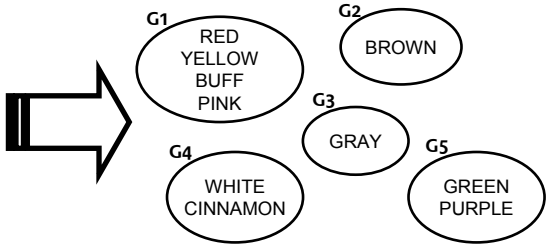

FIG 5 : Groupement de valeurs MODL de la variable couleur de chapeau pour la classification de la base Mushroom en deux classes.

La FIG 5 illustre le résultat du groupement des valeurs de la variable couleur de chapeau grâce à l'optimisation du critère pour la classification des champignons comestibles et vénéneux (Blake et Merz, 1996).

La discrétisation et le groupement de valeurs optimaux sont recherchés en optimisant les critères d'évaluation, au moyen de l'heuristique gloutonne ascendante décrite dans (Boullé, 2005). La complexité algorithmique en O (JN log (N)) de cette heuristique associée à l'excellente fiabilité de la méthode nous permet de traiter un grand nombre de variables, de l'ordre de 50 000. Un modèle bayesien naïf sélectif (Boulle 2007), basé sur une sélection régularisée des variables et une moyenne de modèles, est ensuite construit pour supprimer les variables redondantes et pour produire les scores.

3 Un déploiement efficace

3.1 Le principe

Pour faciliter le déploiement d'un modèle, nous proposons d'extraire du data folder, une table de parangons. La table des parangons contient les individus représentatifs des variables explicatives utilisées par le modèle. Les parangons sont reliés par un index à toute la population. Les scores produits par l'application du modèle sur la table des parangons sont déployés sur toute la population par une simple jointure entre la table des parangons et l'index. Chaque instance de l'entrepôt de données se voit ainsi attribuer le score de son parangon. Cette méthode de déploiement est particulièrement efficace lorsque le modèle à déployer est récurent. Par exemple pour des campagnes marketing mensuelles, seule la table réduite des parangons est construite chaque mois pour produire le score de toute la population. Cette approche permet d'augmenter considérablement le nombre de scores récurrents pouvant être produits sur une même architecture technique.

3.2 La sélection des parangons

La table des parangons est déterminante pour la performance finale du système. Une table de parangons peu représentative pourrait conduire à la construction de scores inefficace sur l'ensemble de la population. A contrario, une base de parangons de très grande taille diminuerait sensiblement l'intérêt de l'utilisation des parangons. Nous devrons donc gérer au mieux le compromis entre la réduction de volumétrie et la représentativité de la base.

Pour sélectionner efficacement les instances, notre approche est d'utiliser un algorithme d'échantillonnage en une seule passe optimisant un critère de représentativité de l'échantillon. L'algorithme Reservoir Sampling permet de construire un échantillon grâce à un réservoir maintenu en ligne en ajoutant et supprimant aléatoirement des instances du réservoir (Vitter 1985). Cet algorithme stochastique est très bien adapté au cas des flux de données de taille infinie et pas nécessairement stationnaires. Néanmoins le problème que nous cherchons à résoudre est de nature différente : nous connaissons la table de données que nous voulons échantillonner. Dans ce cas un algorithme déterministe est plus adapté. Il permet d'optimiser un critère de représentativité de l'échantillon (Li 2002) en un nombre connu à l'avance d'itérations. Des versions déterministes de Reservoir Sampling existent. L'algorithme Deterministic Reservoir Sampling (Akcan et al 2006) minimise sur le réservoir un critère local inspiré de l'algorithme FAST (Chen et al 2002) : la distance L2 entre l'échantillon et l'ensemble total est minimisée en ajoutant et supprimant en ligne des instances au réservoir.

Afin de réduire le temps de traitement, pour optimiser le critère local nous nous contentons de remplir au fur et à mesure un réservoir jusqu'à ce qu'il atteigne la taille P désirée sans supprimer d'instances. Le risque théorique est de tomber dans un minimum local. En pratique, la taille de notre échantillon est suffisamment importante tant en termes de proportion que de nombre d'instances pour que ce risque soit faible (de l'ordre de 1% pour une taille de 10000). L'algorithme utilisé est le suivant :

1. Le réservoir est initialisé par les K premières instances rencontrées.
2. Pour p allant de K à P :
   - une instance est choisie dans une fenêtre de recherche de taille M de manière à minimiser C(p) le critère de qualité de l'échantillon,

- la fenêtre est ensuite décalée de L instances de manière à obtenir un échantillon de taille P lorsque la table complète de taille N sera parcourue, avec L = (N-M)/P.

La taille de la fenêtre de recherche permet de régler un compromis entre le coût de traitement et la précision de l'algorithme : plus M est grand, mois l'algorithme est rapide, mais plus il est précis. Le paramètre K permet d'initialiser l'algorithme d'optimisation. Une trop petite valeur de K risque de rendre totalement inefficace les premières itérations de l'optimisation du Khi².

Nous utilisons le critère du Khi² pour mesurer la proximité entre l'échantillon et la table complète. L'algorithme de discrétisation et groupement de valeurs, décrit à la section précédente, nous permet d'extraire une représentation binaire : chaque variable binaire i correspond à la fréquence d'une modalité d'une variable discrétisée. Le critère à minimiser s'écrit alors :

$$C(p) = \sum_{i=1}^{m} \frac{(pS_i^p - pS_i)^2}{pS_i}$$

Où S_i est la fréquence théorique de la variable binaire i donnée par l'algorithme de discrétisation et groupement de valeurs, et S_i^p la fréquence observée dans l'échantillon de taille p.

3.3 L'indexation

Le problème qu'on cherche à résoudre est simple à énoncer: étant donné un individu, trouver son plus proche voisin parmi la base des parangons; on souhaite faire cela pour tous les individus de l'entrepôt.

La recherche du plus proche voisin est une opération coûteuse. Son implémentation naïve implique une recherche exhaustive parmi les parangons, donc une complexité en O(n.m.p), n étant le nombre d'individus dont on cherche les voisins, m le nombre de variables dans l'espace de représentation et p le nombre de parangons. Afin d'accélérer la recherche du plus proche voisin, on peut être amené à préférer un compromis entre vitesse et performance plutôt que de viser la performance maximale (trouver le plus proche voisin). C'est précisément ce que permet l'algorithme Locality Sensitive Hashing (Gionis et al 1999). Il repose sur une technique de hachage pour sélectionner de bons candidats parmi les parangons pour être proche voisin de l'instance considérée. On applique ensuite une recherche exhaustive parmi ces candidats. Notre implémentation de cette technique permet de ramener la complexité de la recherche du plus proche voisin à $o\left(n.m.\sqrt{p}\right)$ soit un gain d'un facteur 300 pour 100 000 parangons, et laisse à l'utilisateur le contrôle du compromis vitesse / performance.

4 Expérimentations

Pour mesurer la fiabilité des scores produits par notre approche, nous avons comparé des scores construits pour les campagnes marketing de France Télécom avec ou sans notre technologie. Nous avons alimenté la plateforme avec différents jeux de données provenant des applications décisionnelles du groupe France Télécom. Nous avons consolidé des informations de 1 000 000 de clients du groupe sur un passé récent entre janvier et juin 2005.

Les quatre premiers mois ont été utilisés pour construire les profils des clients et les deux derniers pour calculer la variable cible. 20% des clients sont réservés pour l'évaluation les modèles. Pour évaluer la qualité d'un modèle, nous utilisons une courbe de gain (FIG 6). Ce type de courbe permet de sélectionner un modèle par rapport à son efficacité économique. L'axe des abscisses correspond à la proportion de la population visée par les courriers et donc au coût de la campagne. L'axe des ordonnées identifie le pourcentage de la population cible touchée, et donc le gain de la campagne marketing.

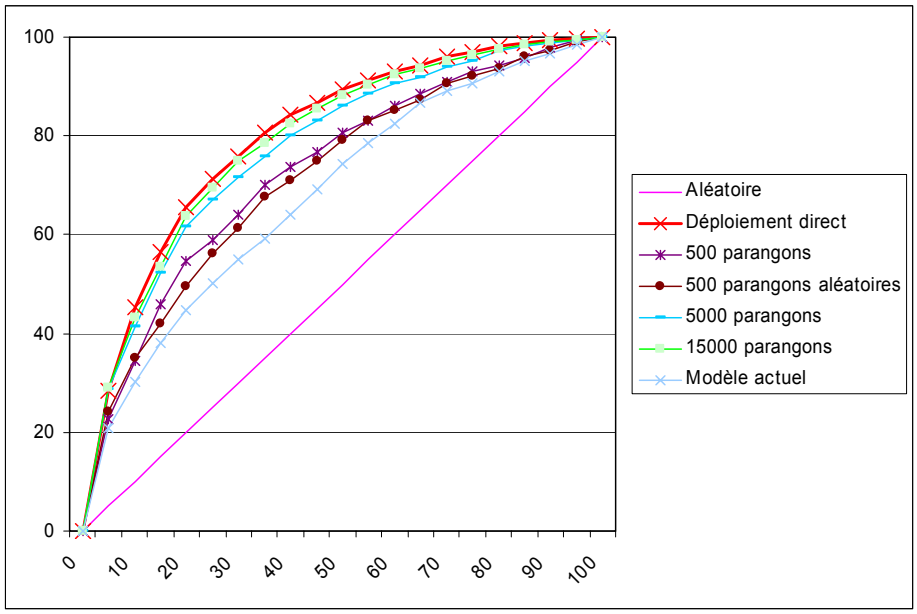

FIG 6 : Courbe de gain des différents modèles pour la résiliation ADSL

Sur la FIG 6, les courbes de gain de différents modèles de prévision de résiliation de l'abonnement à l'ADSL sont tracées. La diagonale représente la performance du modèle aléatoire. Si nous contactons 20% de la population par ce moyen, nous toucherons 20% des clients qui vont rendre leur abonnement dans les deux prochains mois.

Le modèle utilisé actuellement par les services marketing de France Télécom se base sur 200 variables explicatives. Lorsque 20% de la population est contactée, 45% des clients fragiles sont contactés. Le gain est de + 25% par rapport au ciblage aléatoire.

L'automatisation de la recherche de représentation, nous a conduit à sélectionner un modèle se basant sur 191 variables explicatives choisies dans un ensemble de 50 000 variables. Le modèle est ensuite déployé sur toutes les instances en utilisant un nombre variable de parangons : 500, 5 000, 15 000, et déploiement direct sur la population. Avec un déploiement direct sur toutes les instances, lorsque 20% de la population est contactée en se basant sur ce ciblage, 65% des clients qui vont rendre leur abonnement dans les deux prochains mois sont touchés. Par rapport à la technique actuelle, pour le même nombre de courriers, 20% supplémentaire de la population cible est touchée. Cette amélioration du ciblage est vraie sur toute la courbe de gain.

Lorsque la technique de déploiement des scores par les parangons est utilisée, il y a une perte potentielle de fiabilité qui dépend du nombre de parangons utilisés. Plus ce nombre est important, plus le ciblage est proche du meilleur possible, mais plus il est couteux à utiliser. Lorsque 5 000 parangons sont utilisés pour représenter 1 000 000 clients, à 20% de la population, 60% des clients fragiles sont touchés. Le gain reste de + 40% par rapport à l'aléatoire et de + 15% par rapport à la technique actuelle. Avec 15 000 parangons, les performances obtenues sont quasiment similaires à celles obtenues avec un déploiement direct. Pour évaluer la qualité de l'algorithme de sélection de parangons, nous avons comparé les performances obtenues lorsque les parangons sont sélectionnés de manière aléatoire, avec les performances obtenues lorsque les parangons sont sélectionnés en optimisant le Khi² entre la distribution théorique des variables et celle obtenue dans l'échantillon. Avec 500 parangons, à 20 % de la population, 50 % de la cible est atteinte pour la sélection aléatoire contre 55 % pour l'optimisation locale du Khi² (FIG 6).

Le processus complet d'extraction de la table des parangons à partir d'un million de clients et d'un espace de recherche de 50 000 variables correspond à 20 heures de calcul sur un serveur comprenant quatre processeurs à 3 Ghz muni chacun de 3 Go de mémoire. Deux tiers du temps de traitement correspond à la sélection de la représentation et un tiers par la recherche et l'indexation des parangons. Une fois les parangons obtenus, la production des scores à partir de la table des parangons est faite en moins d'une minute. Lorsque le déploiement direct est utilisé, 2 heures de traitements sont nécessaires pour générer une table d'un million d'instances caractérisées par 191 variables explicatives et appliquer le modèle sur cette table. L'utilisation de parangons est efficace pour déployer un score récurrent, comme les scores de fragilité ou de recrutement ADSL. Pour un score opportuniste, comme l'appétence à une offre particulière, le déploiement direct est plus efficace.

5 Conclusion

Nous avons décrit une plateforme de fouille de données permettant de construire des modèles de prévision basés sur un nombre de variables explicatives de deux ordres de grandeurs au-dessus de ce qui se fait actuellement. La conséquence est une nette augmentation de la qualité des modèles. Cette plateforme repose sur une architecture novatrice permettant d'automatiser les traitements couplée avec des méthodes performantes de construction, sélection, et indexation de variables et / ou d'instances.

Le temps de traitement du à la mise à plat des données reste la principale limite à l'exploration d'un espace de recherche plus grand. Pour aller plus loin dans l'exploration des grandes masses d'information, nous devrons élaborer une méthode de parcours de l'espace des variables permettant de se diriger plus rapidement vers les zones contenant les variables pertinentes.

Références

Akcan H., A. Astashyn, H. Brönnimann, L. Bukhman (2006). *Sampling Multi-Dimensional Data,* Technical Report TR-CIS-2006-01, CIS Department, Polytechnic University, 2006.

Blake, C.L. et C.J. Merz (1996). *UCI Repository of machine learning databases*. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Boullé, M. (2005). *A Bayes optimal approach for partitioning the values of categorical attributes*. Journal of Machine Learning Research, 6:1431-1452

Boullé, M (2006). *MODL: a Bayes optimal discretization method for continuous attributes*. Machine Learning, 65:131-165.

Boullé, M. (2007). *Compression Based Averaging of Selective Naïve Bayes Classifiers*, Journal of Machine Learning Research, 8:1659-1685.

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, et R. Wirth (2000). *CRISP-DM 1.0: step-by-step datamining guide*.

Chen B., P. J. Haas, P. Scheuermann. (2002). *A new two-phase sampling based algorithm for discovering association rules*. ACM SIGKDD. pp. 462-468.

Fayyad U. M., G. Piatetsky-Shapiro, et P. Smyth (1996) *From data mining to knowledge discovery : an overview*. Advances in Knowledge Discovery and Data Mining, 1-34, MIT Press.

Féraud R., F. Clérot (2001), *A methodology to explain neural network classification*, Neural Networks, 15:237-246.

Gionis A., P. Indyk, R. Motwani (1999). *Similarity Search in High Dimensions via Hashing*. VLDB Conference.

Guyon, I., A. Elisseeff (2003), *An Introduction to Variable and Feature Selection*. Journal of Machine Learning Research, 3:1157-1182.

Guyon, I., S. Gunn, M. Nikravesh, L. Zadeh (2006), Feature Extraction and Applications, Springer.

Kohavi, R. et G. John (1997). *Wrappers for feature selection.* Artificial Intelligence, 97(1-2):273-324.

Lemaire V., R. Féraud (2006), *Driven forward features selection: a comparative study on neural networks*, In ICONIP, Hong-Kong, 693-702.

Li X. *Data reduction via adaptive sampling*. Communication in Information and Systems, Vol. 2, No. 1, pp. 53-68, Juin 2002.

Pyle, D (1999). *Data Preparation for Data Mining*. Morgan Kaufmann.

Saporta, G. (1990). *Probabilités analyse des données et statistique*. Editions TECHNIP.

Vitter J. S. (1985). *Random sampling with a reservoir*. ACM Trans. Math. Software, 11(1):37-57.

Zighed, D. A. et R. Rakotomalala (2000). *Graphes d'induction*. Hermes.