Table des matières

Table des figures

Liste des tableaux

Glossaire

Acc : Accuracy.
AdaBoost : Adaptative Boosting.
AdaCost : cost-sensitive adaptative Boosting.
Bagging : Bootsrap aggregation.
FP : Faux Positive.
FN : Faux Négative
FURIA : Fuzzy Unordered Rule Induction Algorithm.
gg-SMOTE : Gabriel-graph-based SMOTE.
GLMBoost : Generalized Linear Model Boosting.
KEEL : Knowledge Extraction based en Evolutionary Learning.
KNN : K-Nearest Neighbors.
LS-SVM : Least-Square SVM.
MLP : Multi-Layer Perceptron.
MSMOTE : Modified Synthetic Minority Over-sampling Technique.
PSO : Particle swarm optimization.
RBF : Radial Basic Function.
RF : Random Forest.
RSM : Random subspace method.
RUS : Random Under Sampling.
RUSBoost : Random Under Sampling Adaptative Boosting.
SBC : under Sampling Based Clustering.
Sen : Sensibilté.
SMOTE : Synthetic Minority Over-sampling Technique.
SMOTEBoost : Synthetic Minority Over-sampling Technique Adaptative Boosting.
Spe :Spécificité.
SVM : Séparateurs à Vaste Marge.
SUNDO : Similarity based UnderSampling and Normal Distribution based Over-sampling.
VP : Vrai Positive.
VN : Vrai Négative.
WS+MSS : Wilson's editing+ Modified selective subset

Introduction générale

Ces derniers temps, l'expression « intelligence artificielle » est fréquemment utilisé par le public car il s'agit d'un domaine en constante évolution notamment grâce aux progrès des technologies informatiques et entre autres grâce aux capacités toujours plus grandes des machines pour effectuer les calculs.

Depuis ses origines, l'intelligence artificielle avait généralement pour objectif de doter les machines de la capacité à pouvoir effectuer des tâches réputées "intelligentes" tendant à rendre la machine capable d'acquérir de l'information, de raisonner sur une situation statique ou dynamique, de résoudre des problèmes combinatoires, de faire un diagnostic, de proposer une décision, un plan d'action et d'expliquer.

C'est dans ce cadre plus ou moins précis qu'apparaît l'apprentissage automatique qui constitue l'un des sous-domaines de l'intelligence artificielle les plus prometteurs de cette dernière décennie. Mais comme tout domaine de recherche, plusieurs problèmes peuvent interviennent le plus connu est l'influence de la qualité de données sur le bon apprentissage comme le représentent clairement les données déséquilibrées.

Les données sont dites déséquilibrées lorsque au moins une classe est sous représenté par rapport aux autres, la classe d'intérêt est généralement la classe minoritaire ce qui perturbe les algorithmes d'apprentissage et devient incapable reconnaître et classer les instances minoritaires, le déséquilibre de classe peut se manifester dans des problèmes du monde réel tels que la détection de fraude, la catégorisation de textes, ainsi que le diagnostic médical qui représente le domaine le plus sensible et lors d'une erreur la vie d'un patient est en jeu.

Ce problème est devenu un défi dans la communauté de data mining, car il est présent dans de nombreuses applications du monde réel. En raison de l'importance de cette question, une grande quantité de techniques ont été développées peuvent se diviser en deux grandes familles :

1. Modifications au niveau des données par les différents méthodes d'échantillonnages
2. Modifications au niveau des algorithmes d'apprentissage comme les méthodes d'ensemble, méthodes sensibles au coût

Nous nous intéressons dans ce projet aux méthodes d'ensemble. Leur approche est largement utilisée dans la littérature et elle a donné de très bons résultats dans

le cadre de déséquilibre de classe. Grâce au principe de combinaison de plusieurs classifieurs chacun donne un résultat différent de l'autre, de ce fait, nous exploitons tout l'espace de solution.

Ce mémoire de fin d'études de Master IBM est constitué de 4 chapitres qui sont représentés comme suit :

- Chapitre 1 : Contexte de la distribution déséquilibrée des données, englobe des définitions sur données déséquilibrées et la problématique abordée
- Chapitre 2 : État de l'art, expose les travaux du domaine de déséquilibre de données
- Chapitre 3 : Matériel et méthodes, détaille les différents techniques utilisées dans ce projet
- Chapitre 4 : Expérimentations et résultats, dévoile l'étude expérimentales

En dernier lieu, une conclusion générale qui résume les points essentiels de ce travail.

Chapitre 1

Contexte de la distribution déséquilibrée des données

1 Introduction

Récemment, le problème de déséquilibre de classes a attiré l'attention de plusieurs chercheurs, cela dus à l'importance de l'équilibrage de données pour la création d'un bon classifieur.

La plupart des données réelles sont déséquilibrées, le nombre d'une classe est largement grand par rapport à l'autre, c'est spécialement le cas pour le diagnostic médical, où les personnes malades sont toujours rares (classe minoritaire) en les comparent avec les non-malades (classe majoritaire). De plus, lors de l'émergence d'une nouvelle maladie, elle est d'un impact amoindri à ses débuts, ses causes sont inconnues et ses symptômes peuvent être communs avec ceux des autres maladies déjà existantes, par suite, il est difficile de la reconnaître et peut être facilement assimilée à ces dernières.

Dans ce chapitre, nous allons exposer le problème d'apprentissage en distribution déséquilibrée à fin de proposer une solution.

2 Définition

La distribution des instances de classes joue un rôle très important pour atteindre une bonne classification.

Une base de données est dite déséquilibrée quand le nombre des instances d'une classe est petit par rapport aux autres classes (voir figure 1.1) [BAH10].

De manière plus simple, le nombre d'instances négatives (majoritaires) est largement supérieur au nombre d'instances positives (minoritaires).

Dans le domaine médical, le médecin s'intéresse au cas minoritaire aux quels il n'a pas beaucoup d'exemple pour l'étude approfondie.

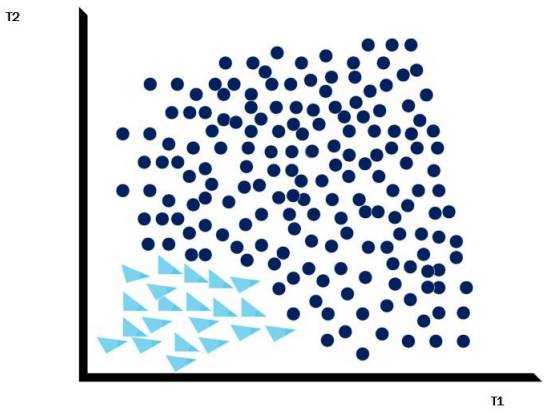

FIGURE 1.1 – Exemple de distributions de données déséquilibrées

3 Notions de déséquilibre et d'asymétrie

Dès l'utilisation des données réelles en fouille de données, on est généralement en présence d'un très grand nombre d'individus, d'un très grand nombre de descripteurs, ou encore de données manquantes, éparses, ou bruitées. Un problème de plus en plus considéré par la communauté scientifique depuis quelques années : ce problème est l'asymétrie des classes.

Or la plupart des problèmes industriels concernent les classes distribuées de manière asymétrique. En apprentissage supervisé, l'asymétrie peut être en deux formes principales : le déséquilibre des classes, et l'asymétrie des coûts. Le déséquilibre de classes concerne les problèmes où l'une des modalités de la variable cible est beaucoup moins représentée que les autres, ce qui perturbe les algorithmes d'apprentissage. Ce problème est souvent rencontré dans les problèmes de diagnostic médical. L'asymétrie des coûts concerne les cas où les coûts des erreurs ne sont pas symétriques [Mar08].

L'asymétrie est devenue un défi majeur de l'apprentissage supervisé, le déséquilibre de données pouvant atteindre 1 pour 100, 1 pour 1000, 1 pour 10000 et souvent encore plus. Comme le notent Verhein & Chawla [VC06] dans des applications comme le diagnostic médical ou la détection de fraudes, les jeux de données déséquilibrés sont la norme et non l'exception [HAF13].

Dans le cas de déséquilibre, il est difficile de répondre à la première hypothèse, si 99% des données appartient à une seule classe, il sera difficile de faire mieux que 1% d'erreur obtenu en classant tous les individus dans cette classe Weiss [Wei04] propose de distinguer plus précisément les différents problèmes de déséquilibre :

1. Métrique inapproprié : Les mesures utilisées au cous du processus d'apprentissage ne sont pas adaptées au classes déséquilibrées
2. Manque absolu de données : c'est le problème principal du déséquilibre, le nombre d'instances d'une classe est peut nombreux (rare) pour représenté un

concept

3. Manque relatif de données : Les objets d'une classe ne sont pas rares au sens absolu mais moins représenté par rapport au autre classe
4. Données bruité (figure1.2(a)) : Le bruit a plus d'impact sur les classe rare que sur les classe fréquente
5. Petites disjoint : (figure1.2(b)) la classe minoritaire est divisée en petit ensembles séparés

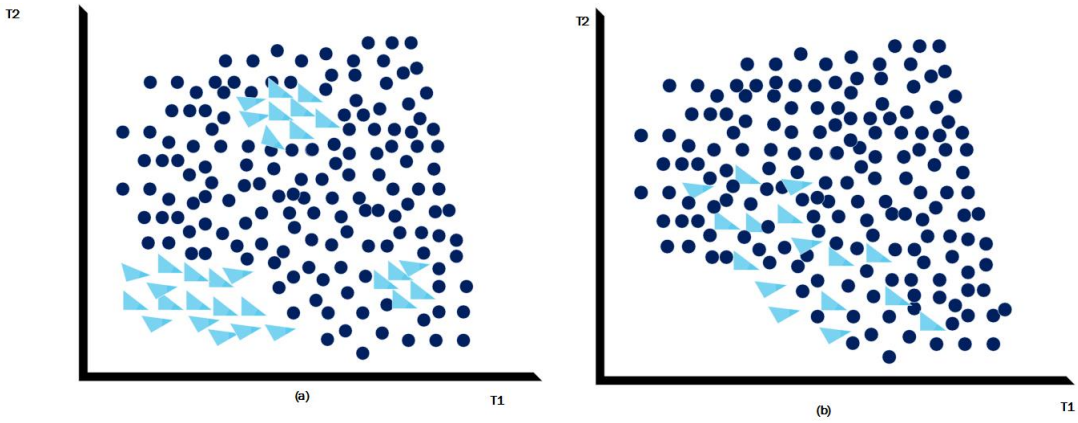

FIGURE 1.2 – Exemple de difficulté en données déséquilibrées ... (a) petites disjointes (b) séparabilité de classe

4 Problématique

Une base de données qui a une distribution inégale entre classes est considérée comme une base de données déséquilibrée. Tandis que la classe avec moins d'instances est toujours la plus importante.

Afin de mettre en évidence le problème d'apprentissage des données déséquilibrées dans le domaine médical, nous présentons un exemple avec la base de données des images mammographiques : une collection d'images acquise à partir des séries d'examens de différents patients.

La classe est ''positive" ou ''négative" pour un cas ''cancéreux" ou ''non cancéreux" respectivement.

Le nombre de cas non-cancéreux dépasse largement le nombre de cas cancéreux. En effet, cette base contient 10.923 ''négative" (classe majoritaire), et 260 ''positive" (classe minoritaire).

On a besoin d'un classifieur qui nous donne une bonne reconnaissance pour les 2 classes minoritaire et majoritaire. Concrètement, les classifieurs existant atteignent la plus haute précision avec la classe majoritaire, par contre avec la classe minoritaire, ils ne dépassent pas les 0-10% de précision (voir Figure 1.3).

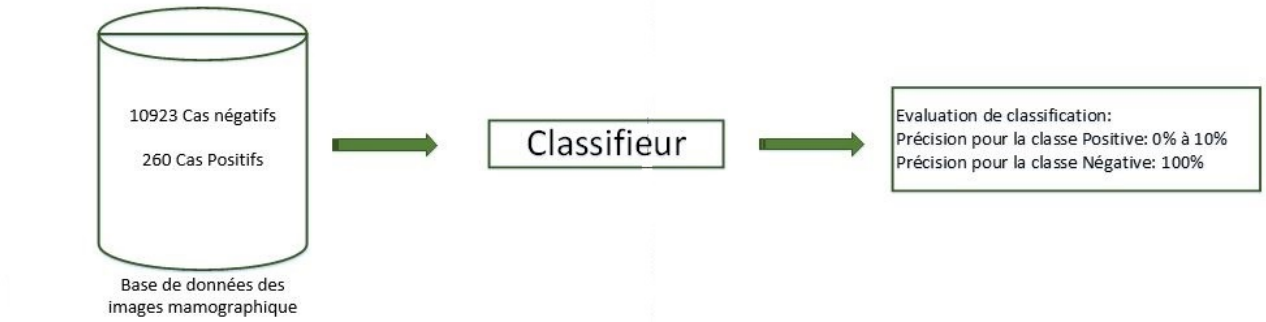

FIGURE 1.3 – Le problème de déséquilibre des données

Supposons que le classifieur atteint 10% de précision de la classe minoritaire, analytiquement cela signifie que 234 de la classe minoritaire sont mal classé la conséquence de cela est équivalent au 234 cas cancéreux classés non-cancéreux ce résultat peut être coûteux plus que de classer non cancéreux comme cas cancéreux [GFT+12].

De ce fait, il est primordial de mettre en place de nouveaux modèles qui s'adaptent au déséquilibre tout en minimisant l'erreur, afin de construire un système d'aide au diagnostic médical précis et robuste.

5 Motivations

Dans la communauté de data mining, la classification des données souffrent de la distribution déséquilibrée des classes. Ce problème survient lorsque le nombre d'exemples qui représentent une classe est beaucoup plus bas que ceux des autres classes. Sa présence dans de nombreuses applications dans le monde réel a apporté une croissance de l'attention des chercheurs et ils ont proposé plusieurs méthodes.

Ces dernières années, un grand nombre de travaux en apprentissage automatique ont porté sur les méthodes d'agrégation de classifieurs. Ces méthodes consistent à induire nom pas un, mais plusieurs classifieurs puis dans une seconde phase à utiliser un mécanisme d'agrégation des classifieurs pour produire une fonction de décision h unique.

Les méthodes ensembliste ont beaucoup contribué à amélioré l'efficacité des procédures de prédiction en data mining en particulier le boosting qui a la particularité d'être adaptatif au sens où il est forcé à se spécialiser sur les exemples difficile à prédire comme notre cas de données déséquilibrées.

Ces méthodes d'agrégation sont efficaces d'un point de vue biais variance, mais aussi grâce aux trois raisons fondamentales expliquer par Dietterich [Die00] :

Raison statistique : un algorithme peut être considéré comme une recherche d'un espace H des hypothèses pour identifier la meilleure hypothèse dans l'espace le problème statistique surgi quand la quantité de données d'apprentissage disponible est très petites comparé à la taille de l'espace d'hypothèses sans données suffisante, l'algorithme d'apprentissage peut trouver plusieurs hypothèses dans H donnant la même exactitude sur les données d'apprentissage en construisant un ensemble, en se basant sur tous classifieurs l'algorithme peut faire la moyenne de leurs votes et réduire le risque de choisir le mauvais classifieur.

Raison informatique : différents algorithmes d'apprentissage fonctionnent en exécutant une certaine recherche locale qui peut se coincer dans des optimums locaux. Un ensemble construit en exécutant la recherche locale de différents points de départ peut fournir une meilleure approximation de la fonction cible inconnue que n'importe quels différents classifieurs.

Raison de présentation : dans la plupart des applications d'apprentissage, la fonction cible F ne peut être représenté par aucune des hypothèse constituant H en formant des somme pondéré d'hypothèses de l'ensemble H.

Par conséquent dans notre projet de fin d'études, nous avons opté pour le traitement des données déséquilibré par les méthodes d'ensemble.

6 Conclusion

Dans ce chapitre, nous avons exposé le contexte des données déséquilibrées, leur problématique particulière au domaine médical, en mettant en évidence l'importance cruciale d'avoir un bon diagnostic dans cette application très sensible qui n'admet pas d'erreurs.

Les données déséquilibrées ont besoin d'un traitement particulier, la majorité des algorithmes ne s'adaptent pas avec la classe minoritaire, ils la considèrent comme du bruit malgré que c'est l'information cible. C'est pour ces raisons, plusieurs chercheurs se sont intéressés à ce contexte et cette problématique de distribution déséquilibrée, plusieurs travaux ont vu le jour dans ce sens, nous allons les voir en détails dans le chapitre suivant.

Chapitre 2

État de l'art

1 Introduction

La plupart des méthodes de classifications existantes ne s'adaptent pas avec la classe minoritaire quand la classe est extrêmement déséquilibrée.

Ce problème est devenu un défi dans la communauté de data mining, car il est présent dans de nombreuses applications du monde réel. En raison de l'importance de cette question, une grande quantité de techniques ont été développées.

Ces techniques peuvent se diviser en 2 catégories :

1. Par modification au niveau des données, cette approche rendre la base de données déséquilibrée équilibrée à travers les méthodes d'échantillonnages.

2. Par modification au niveau des algorithmes d'apprentissage, par l'amélioration des algorithmes traditionnelle et les adaptés au déséquilibre de classe.

2 Approches par modification au niveau des données

Le changement de la distribution de classe est effectué au niveau de la base d'apprentissage permet de construire la base et de faire une nouvelle base équilibrée.

La base de données peut être équilibrée par le sous-échantillonnage de la classe majoritaire, et le sur-échantillonnage de la classe minoritaire ou les deux techniques au même temps (voir Figure 2.1).

Les études ont montré qu'une base équilibrée améliore les performances de classification par rapport à une base déséquilibrée.

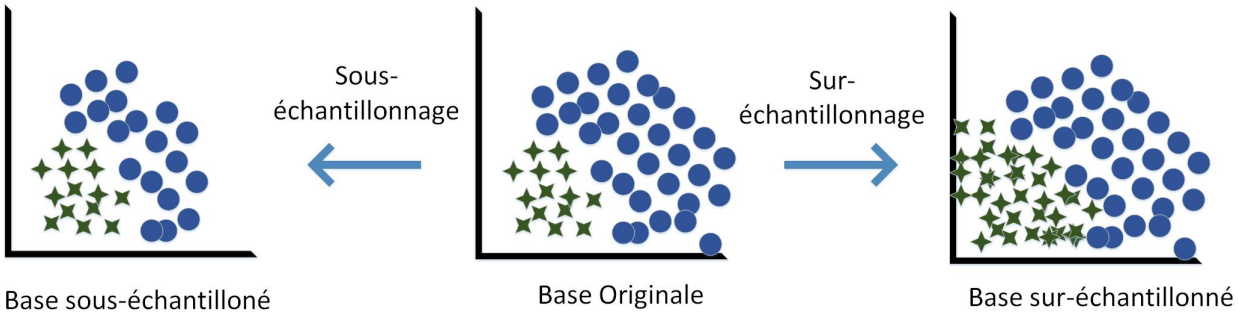

FIGURE 2.1 – Le principe d'échantillonnage

Parmi les solutions proposées est l'ajout d'individus aléatoirement [CJK04], mais un risque de sur-apprentissage peut survenir parce que la taille de la base va augmenter. Pour éviter ce risque, Chawla et al. [CBHK02] ont proposé une nouvelle technique (Synthetic minority over sampling thechnic SMOTE) qui permet de générer des individus artificiels dans la classe minoritaire. Pour chaque individu de la classe minoritaire, ses k plus proches voisins appartenant à la même classe sont calculés, puis un certain nombre d'entre eux (selon le taux de sur-échantillonnage voulu) sont sélectionnés. Des individus artificiels sont ensuite disséminés aléatoirement le long de la ligne entre l'individu de la classe minoritaire et ses voisins sélectionnés.

Il existe un autre type d'échantillonnage est sous échantillonnage de la classe majoritaire plusieurs études sont focalisé aussi sur ce type.

La méthode la plus simple consiste à supprimer aléatoirement [BPM04] des individus appartenant à la classe majoritaire, de manière à rééquilibrer le jeu de données. Cette méthode a l'avantage d'être très simple à mettre en œuvre, mais elle risque de supprimer des données d'apprentissage des individus importants pour le concept de la classe majoritaire.

Le tableau Table 4.1, ci-dessus représente quelque travaux de ré-échantillonnage de base de données déséquilibré

Auteurs	titres	Méthodes et expériences	Résultats
Lee et al. 2010 [LYCL10]	A hybrid algorithm appliesd to classify unbalanced data	Ce document présente l'hybridation de 3 types d'algorithmes : sur-échantillonnage aléatoire, arbre de décision, optimisation par essaims particulaire (PSO). Premièrement, ils ont traité la base par l'utilisation de sur-échantillonnage aléatoire, ensuite PSO et arbre de décision pour sélectionner les caractéristiques intéressantes et classées la base de données. Cette méthode a été comparée par l'utilisation d'autres classifieurs KNN, SVM, arbre de décision.	La meilleure précision obtenue est 99,5% réalisé par la méthode proposée sur la base de données zoo.
Gao et al .2011 [GHCH11]	A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems	Cet article combine des outils de résolution du déséquilibre comme SMOTE et un algorithme d'optimisation essaims particulaire (osp) par un classifieur de type réseau de neurones dont la fonction d'activation est la fonction radiale (RBF). Les bases de données utilisées sont : Pima, haberman, ADI. Ils ont créé différents changement du taux de sur-échantillonnage sur les bases d'apprentissages 0%, 100%, 300%, 500%, 800%, 1000%, 1500%, et 2000%. La méthode proposée a été comparée avec le classifieur Knn	Les résultats expérimentaux pour cette étude ont démontré que la technique proposée est meilleure grâce à son principe d'équilibrage et de minimisation d'erreurs
Garcia et al. 2012 [GSM12]	On the effectiveness of preprocessing methods when dealing with different levels of class imbalance	L'article présente une méthode qui permet de déterminer l'influence du degré de déséquilibre sur les techniques d'échantillonnage, utilisant différents algorithmes d'apprentissages sur 17 bases de données. Ils ont divisé l'ensemble de bases en 2 catégories, les bases avec un déséquilibre fort et bases avec déséquilibre faible. Les classifieurs sont appliqué sur les données pré-traitées par les différentes stratégies d'échantillonnage. Les méthodes d'échantillonnage choisies sont : RUS, WE+MSS, SMOTE, et gg-SMOTE. Les approches sont : KNN, SVM, réseaux bayesiens, arbre de décision et RBF.	Les résultats des bases de données avec un déséquilibre élevé montrent que le sur-échantillonnage et plus approprié parce qu'il ajoute des instances. Par contre, de l'application du sous-échantillonnage survient une perte d'information. Pour la deuxième catégorie, les deux techniques atteignent des résultats similaires mais toujours mieux que la base originale.

Rahman et al. 2013 [RD13]	Cluster Based Under-Sampling for Unbalanced Cardiovascular Data	Ce papier réalise une étude comparative entre SMOTE et une méthode de sous-échantillonnage basé sur le clustering proposée par Yen et Lee(2009) [YL09]. Les auteurs ont appliqué deux algorithmes d'apprentissage : arbre de décision et FURIA (Fuzzy Unordered Rule Induction Algorithm,)sur la base de données cardiovasculaire.	Le SMOTE montre une bonne classification pour les deux classifieurs mais le sous-échantillonnage est meilleur en terme de temps d'exécution.
Cateni et al. 2014 [CCV14]	A method for resampling imbalanced data sets in binary classification tasks for real-world problems	La méthode proposée est de combiner les deux types d'échantillonnage (sous- échantillonnage, sur-échantillonnage) afin d'obtenir une base de données équilibrée. Cette méthode est nommée "based under sampling and normal distribution based oversampling" (SUNDO). Les auteurs ont comparé SUNDO avec la classification de la base originale sans aucun changement, SMOTE, Under sampling based clustering (SBC) et 4 classifieurs SVM, arbre de décision, réseaux de neurone et réseaux bayesien. L'application est réalisée sur trois bases de données : metal sheet qulity assessment, BreastCancer, occlusion detection	Pour tout les bases de données et tout les classifieurs et pour un même degré de déséquilibre, l'approche atteint des bonnes performances même SMOTE a une haute précision.

TABLE 2.1 – État de l'art des travaux concernant la modification au niveau de données

3 Approches par modification au niveau des algorithmes d'apprentissage

3.1 Apprentissage sensible aux coûts

Le principe de cette catégorie d'algorithmes est de fixer des coûts fixes et inégaux sur les différents types d'erreurs de mauvaise classification. Ces coûts sont représentés généralement par une matrice carrée C de taille $k \times k$, où $C(i, j)$ représente le coût de classer un individu de la classe i vers la classe j. La diagonale principale est généralement nulle : une bonne classification a un coût nul [Mar08].

Une approche plus générale appelée MetaCost proposée par Domingos [Dom99]

est basée sur le re-étiquetage des individus. Domingos propose donc d'estimer pour chaque individu sa classe optimale, i.e. celle qui minimise au final le coût. Pour cela, un ensemble de modèles est généré sur différents échantillons boostrap. Puis par vote, on estime pour chaque individu sa probabilité d'appartenance à chaque classe. Ensuite à chaque individu est affectée la classe qui minimise le coût. Le modèle final est alors construit sur ce jeu de données ré-étiqueté.

3.2 Méthodes d'ensemble

Boosting

Le boosting [FS99] est un algorithme itératif qui consiste à affecter des poids différents aux instances de la base. Après chaque itération, le poids sur les instances mal classées augmente et le poids des instances correctement classées diminue.

Plusieurs techniques de boosting sont proposés pour s'adapter au déséquilibre :

- *Adacost :* Cet algorithme [FSZC] assigne des poids plus élevés pour les erreurs sur la classe minoritaire.
- *RareBoost :* Le principe de cette méthode [JKA01] se résume par la condition suivante : si le nombre de vrais positifs est supérieur au nombre de faux positifs, le poids des individus bien classés diminue, et si le nombre de vrais négatifs et supérieur à celui des faux négatifs le poids des individus mal classés augmente.
- *SMOTEBoost :* constatant que le boosting risque de souffrir de sur-apprentissage en sur-pondérant les individus de la classe minoritaire, Chawla propose l'algorithme SMOTEBoost [SKHN10] qui ajoute des individus artificiels par la méthode SMOTE au lieu de simplement augmenter le poids des individus de la classe minoritaire.

Bagging

Un autre modèle de méthodes d'ensembles proposé par Breiman [BB96], il a fait ses preuves au domaine de déséquilibre, permet de sous échantillonné la base originale en un ensemble de sous-bases et pour chacune on applique un classifieur à la fin de l'apprentissage la sortie est le vote majoritaire de ces ensemble.

Les Forêts Aléatoires"Random Forest"

La forêt aléatoire est la méthode la plus utilisé parmi les méthodes de bagging, c'est un bagging d'arbres de décision, chaque arbre est construit avec un sous-ensemble des variables, tirées aléatoirement [Bre01].

Chen et al. [XLNY09] ont proposé deux méthodes pour utiliser les forêts aléatoires sur les données déséquilibrés :

- *Balanced Random Forest (BRF) :* consiste à effectuer un bootstrap sur la classe minoritaire, puis à tirer le même nombre d'instances dans la classe majoritaire (avec remise). Ainsi, l'échantillon de chaque arbre est équilibré.

– *Weighted Random Forest (WBF) :* consiste simplement à construire des arbres sensibles aux coûts (au niveau du critère de partitionnement avec une pondération de l'indice de Gini, et de l'affectation en modifiant le seuil). Ces deux approches permettent d'obtenir de bons résultats sans complexité supplémentaire par rapport à des forêts aléatoires classiques.

D'autres méthodes basées sur la modification au niveau algorithmique ont proposé dans la littérature sont représenté dans le tableau suivant.

Auteurs	titres	Méthodes et expériences	Résultats
Hoens & Chawla 2010 [HC10]	Generating Diverse Ensembles to Counter The Problem of Class Imbalence	Dans le but d'améliorer la classification sur ce genre de données les auteurs ont travaillés sur plusieurs base de données et plusieurs méthodes individuellement comme : l'AdaboostM1(BT), Bagging(BG), Random Forest(RF), et Random subspace method(RSM), ainsi que deux autres méthodes en combinaison RSM+SMOTE, RSM+Undersumpling. avec un seul classifieur qui est l'arbre de décision C4.5.	Les expérimentations montrent que le meilleur classifieur est RSM+SMOTE, dans 12 des 21 base de données.
Brown & Mues 2011 [BM12]	an experimental comparison of classification algorithms for imbalanced credit scoring	Dans l'étude expérimentale, ils ont choisis 5 base de données financières concernant l'évaluation des risques de clients. Ce travail est une étude comparative entre 10 classifieurs : LS-SVM, Logistic regression, Arbre de décision C4.5, K plus proche voisins, Random forests, Linear and quadratic discriminant analysis, Neural network(MLP), Gradient boosting En modifiant le degré de déséquilibre (30%, 15% , 10%, 5%, 2,5%, 1%).	Le gradiant boosting et random forest sont plus performant pour des données largement déséquilibré, où la performance de LS-SVM diminue par l'augmentation du déséquilibre. Par contre le reste des classifieurs ne sont pas efficace en distribution déséquilibre .
Galar et al. 2012 [GFT+12]	A Review on Ensembles for the Class Imbalance Problem : Bagging-, Boosting and Hybrid Based Approache	Cet article présente un large ensemble de données pour le déséquilibre de classe. Basé sur le bagging et boosting (adaboost, adaboost.M1, adaboost.M2) combiné avec les méthodes d'échantillonnage	Nous devons remarquer la bonne performance des approches telles que RUSBoost ou UnderBagging, qui en dépit d'être des approches simples

		et méthodes sensible au coût (RUSboost, SMOTEBoost, MSMOTEBoost, UnderBagging, SMOTEBagging, MSMOTEBagging, adaC2). L'application a été réalisée avec le logiciel de Keel software et avec toutes les bases déséquilibrées incluent dans ce logiciel.	permettent d'atteindre des performances plus élevées que beaucoup d'autres algorithmes plus complexes. Particulièrement remarquable est la performance de RUSBoost, qui est le moins complexe de calcul parmi les plus performants.
Gahdoum 2013 [HAF13]	Classification des données déséquilibrée médicale	Dans ce projet de fin d'études, l'auteur a travaillé sur la base de données diabète pima. Un algorithme des moindres carrées a été appliqué pour équilibrer les données médicales, ensuite une classification neuronale étant réalisée sur ces données.	Les résultats expérimentaux montrent que la précision de l'algorithme RLS après équilibrage est relié à l'ensemble de données du nombre d'échantillons. L'étude faite nous a montré que l'utilisation de la méthode de moindre carrée est effectivement très pertinente pour la régulation de la base de données du diabète.
Hao et al. 2014 [HWB14]	An efficient algorithm coupled with synthetic minority oversamplingtechnique to classify imbalanced PubChem BioAssay data	Dans ce travail, un algorithme efficace, GLMBoost, couplé avec SMOTE est développé et utilisé pour surmonter le problème pour plusieurs ensembles de données déséquilibrées de PubChem BioAssay. En comparaison avec GLMBoost, Forêt aléatoire (RF) combiné avec SMOTE également adopté pour classer les mêmes ensembles de données.	Les résultats de GLMBoost+SMOTE ne présentent pas de performances élèves, mais démontrent également une plus grande efficacité par rapport SMOTE+RF.

TABLE 2.2 – État de l'art concernant la modification au niveau algorithmique

4 Objectifs de ce travail

Dans ce projet de fin d'études de Master, nous nous focalisons sur la bonne reconnaissance et classification des classes dans une distribution déséquilibrée. L'état de l'art des travaux dans ce sens montrent une grande capacité de discrimination par l'approche boosting.

Le boosting construit un ensemble de classifieurs qui sont ensuite agrégés par une moyenne pondérée des résultats ou un vote. En effet, le boosting construit d'une façon récurrente et itérative l'ensemble des classifieurs chaque classifieur est une version adaptative du précédent en donnant plus le poids aux exemples mal prédits ce procédé permet à l'algorithme de se concentrer sur les exemples les plus difficiles à classifiés, l'agrégation de classifieurs permet au boosting d'échapper au sur-apprentissage, cette méthode réduit à la fois la variance et le biais [BAH10].

Quand on expose le problème du déséquilibre au boosting, ce dernier peut s'adapter au déséquilibre par son principe itératif et correctif, il permet une bonne précision.

L'objectif de notre travail est de vérifier cette théorie et la comparé avec des méthodes récentes afin de surmonter le déséquilibre de classes et de créer un système d'aide au diagnostic médical robuste et précis.

5 Conclusion

Dans ce chapitre, nous avons résumé les différentes approches utilisées dans la littérature. Nous avons remarqué que la méthode de sur-échantillonnage SMOTE a montré son efficacité pour ce problème de déséquilibre due à son principe de création des instances artificielles. De même, les méthodes d'ensembles sont bien adaptées au déséquilibre. De ce fait, dans notre étude, nous nous somme basé sur ces deux techniques.

Chapitre 3

Matériels et méthodes

1 Introduction

L'apprentissage à partir des données déséquilibrées nécessite des stratégies adaptées pour obtenir une classification correcte de la classe minoritaire. Dans la littérature, diverses approches sont proposées pour arriver à une classification plus performante parmi elles les méthodes d'ensemble.

Les méthodes d'ensembles constituent une famille ou un ensemble d'algorithmes qui génèrent une collection de classifieurs et agrègent leurs prédictions.

Dans ce chapitre, nous présentons toutes les informations nécessaires à l'appréhension des méthodes utilisées afin de surmonter le problème du déséquilibre de classes. Ce chapitre est réparti en trois sections :

1. Méthodes d'échantillonnage : sous-échantillonnage et sur-échantillonnage
2. Méthodes d'ensembles : Bagging et Boosting
3. Méthodes hybrides : échantillonnage+méthodes d'ensemble.

2 Méthodes d'échantillonnages

Les techniques d'échantillonnage sont très simples, et facile à programmer, leur objectif principal consiste à rétablir l'équilibre entre les effectifs associés à chaque classe, soit en éliminant le nombre d'exemples majoritaires (sous- échantillonnage), soit en augmentant artificiellement le nombre d'exemples minoritaires (sur-échantillonnage).

2.1 Sur-échantillonnage

Le sur-échantillonnage est un moyen pour rééquilibrer les bases de données, il consiste à augmenter le nombre d'individus appartenant à la classe minoritaire. La première solution est de répliquer aléatoirement des individus. Le risque de cette approche est de ralentir les algorithmes en ajoutant des individus, tout en fournissant des modèles incapables de généraliser (risque de sur-apprentissage) : une règle ayant un bon support dans le jeu d'apprentissage peut en fait porter sur un seul individu

[RD13]. Pour remédier à ces inconvénients, Chawla et al. [CBHK02] ont proposé la méthode SMOTE (Synthetic Minority Oversampling Technique).

Synthétique Minority Oversampling THechnique SMOTE

SMOTE [CBHK02] est une approche de sur-échantillonnage guidé en créant des exemples Synthétiques. Cette approche s'inspire d'une technique qui a fait ses preuves en reconnaissance de caractères manuscrits [HB97].

La classe minoritaire est sur-échantillonnée en prenant chaque exemple de la classe minoritaire. SMOTE introduit les exemples synthétiques au hasard sur la ligne reliant le point de classe minoritaire concerné et un de ses K plus proche voisins. Selon le nombre de sur-échantillonnage requis les voisins des k plus proches voisins sont choisis aléatoirement.

Les exemples synthétiques sont générés de la manière suivante (voir Figure 3.1) :

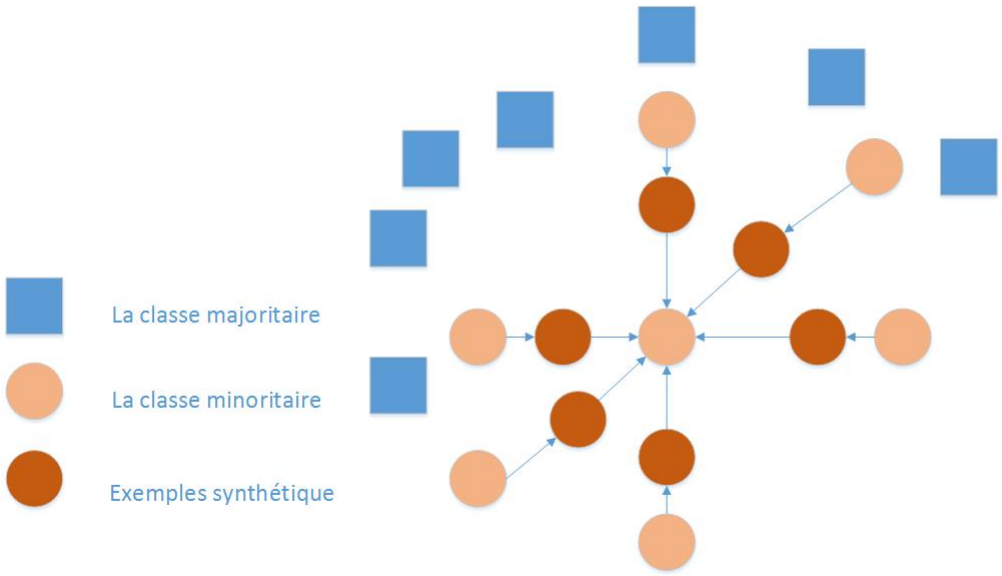

FIGURE 3.1 – Principe de SMOTE

Le principe repose sur la différence entre une instance minoritaire et son voisin le plus proche, cette différence est multipliée par un nombre aléatoire entre 0 et 1, et ajouté à la vectrice caractéristique (voir Figure 3.2), cela provoque la sélection d'un point aléatoire le long du segment de droite entre 2 instances minoritaires (algorithme 1).

Algorithme : SMOTE
Pour *un nombre suffisant d'instance synthetique* **faire**
    Sélectionner une instances minoritaire A
    Séléctionner une des instances les plus proche voisins B
    Sélectionner un poids aléatoire entre 0 et 1 Créer la nouvelle instance synthétique C
    Pour *chaque attributs* **faire**
        attValue_C = attValue_A + (attValue_B - attValue_A) * W
    FinPour
FinPour

Algorithme 1 : pseudo code de l'algorithme SMOTE

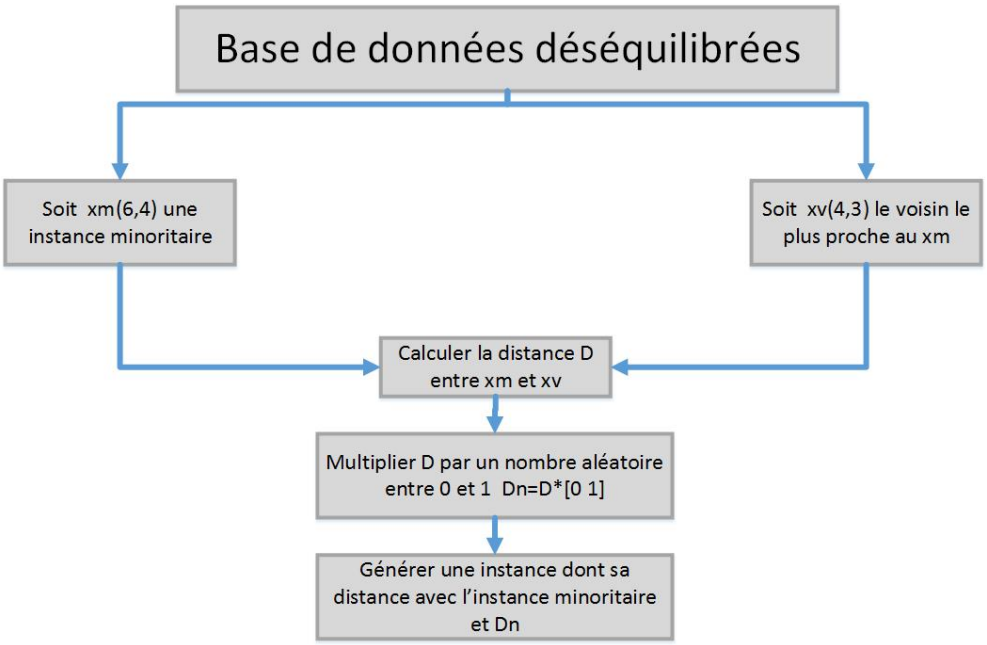

FIGURE 3.2 – Exemple de génération d'un exemple synthétique par l'algorithme SMOTE

2.2 Sous-échantillonnage

C'est la méthode la plus évidente et la plus simple qui consiste à supprimer aléatoirement de la base d'apprentissage des individus appartenant à la classe majoritaire, de manière à rééquilibrer la distribution de classes. Cette approche a l'avantage d'être très simple à mettre en œuvre, mais elle risque de supprimer les individus importants pour le concept de la classe majoritaire [Ber09].

Random UnderSampling RUS

RUS [BPM04] est une approche simple de ré-échantillonnage, les instances de la classe majoritaire sont éliminé au hasard jusqu'à ce que le rapport classe minoritaire sur classe majoritaire est au niveau désiré (voir Figure 3.3).

Théoriquement, l'un des problèmes de sous-échantillonnage aléatoire est que nous ne pouvons pas contrôler les informations de la classe majoritaire qui sont éliminées.

De ce fait, des informations très importantes sur la frontière entre la classe minoritaire et la classe majoritaire peuvent être éliminées, malgré sa simplicité, le sous-échantillonnage aléatoire a été démontré empiriquement pour être une des méthodes de sous-échantillonnage la plus sophistiquée et la plus performante [Liu04].

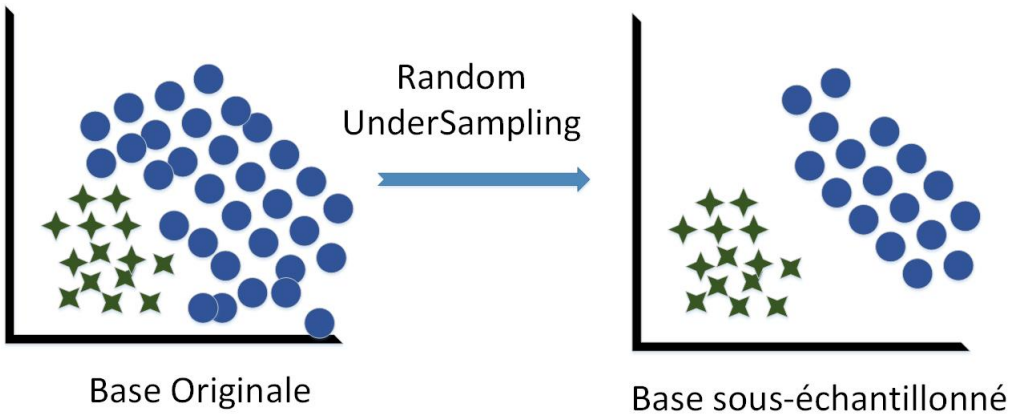

FIGURE 3.3 – Principe de Random UnderSampling

3 Méthodes d'ensemble

Face au très grand nombre de méthodes d'apprentissage statistique présentées dans la littérature, a émergé l'idée de les agréger pour tirer le meilleur parti de leurs avantages respectifs.

Les méthodes d'ensemble sont des algorithmes d'apprentissage qui construisent un ensemble de classifieurs et effectuent une agrégation de leurs prédictions (voir Figure 3.4). Cette idée suit le comportement de la nature humaine qui tend à rechercher plusieurs avis avant de prendre une décision importante (l'union fait la force) [GFT+12].

La principale motivation de combiner des classifieurs est d'améliorer leur capacité de généralisation. Chaque classifieurs fais des erreurs, mais comme ils sont différents, les instances mal classé ne sont pas toujours les mêmes.

Ici, surgit l'idée que les prédicteurs individuels doivent être différents les uns des autres, la majorité ne doit pas faillir pour une même instance. Pour que cela soit possible, il faut également que les prédicteurs individuels soient relativement bons et là où un prédicteur se trompe les autres doivent prendre le relais [GEN10].

Le succès des méthodes d'ensemble se résume en 2 points :

1. Chaque prédicteur individuel doit être relativement bons,

2. Les prédicteurs individuels doivent être différents

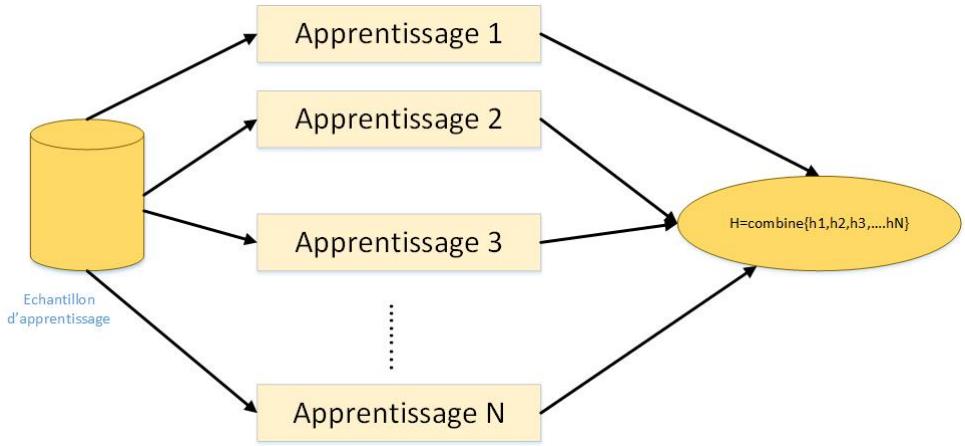

FIGURE 3.4 – Principe général des méthodes d'ensemble

Les méthodes d'ensemble sont divisées selon leur principe de fonctionnement en deux catégories : les méthodes d'ensemble séquentielles qui fonctionnent de manière itérative, et les méthodes d'ensemble parallèles dont l'ensemble des classifieurs fonctionnent d'une façon simultanée.

3.1 Méthodes d'ensemble parallèle

Les méthodes d'ensemble parallèle sont l'agrégation d'un ensemble de classifieurs qui fonctionnent d'une façon simultanée.

Bootstrap Aggregating (Bagging)

Le méthode du Bagging a été introduite par Breimen [BB96]. Le mot bagging est la contraction des mots Bootstrap et Aggregation .

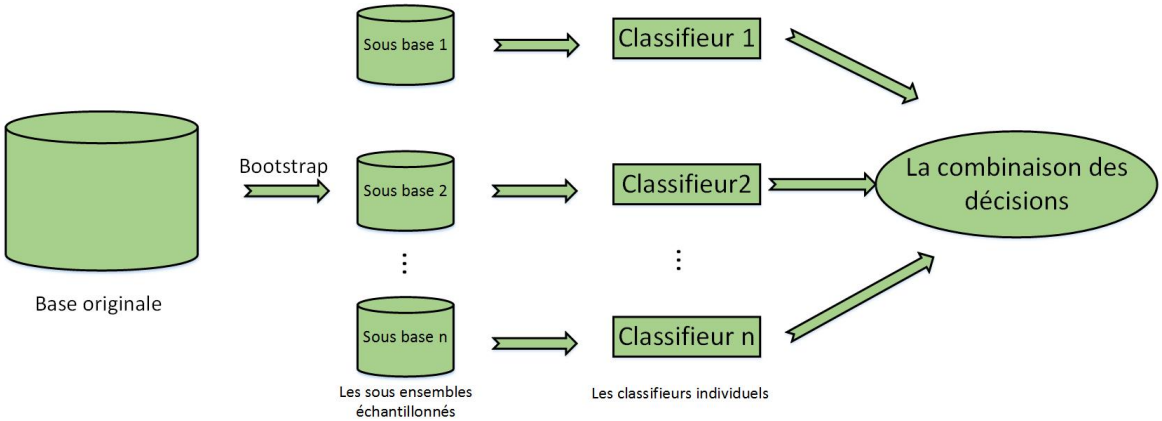

FIGURE 3.5 – Principe de Bagging

Le Bagging repose sur la méthode bootstrap [ET94] pour améliorer la prédiction d'un classifieur. Son principe est donc de tirer uniformément et avec remise les ob-

servations pour créer de nouveaux échantillons, sur chacun desquels est construit un classifieur. La prédiction final est obtenue par vote majoritaire parmi les classifieurs intermédiaires (voir Figure 3.5).

Le Bootstrap est un principe de ré-échantillonnage statistique traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. L'idée du bootstrap est d'utiliser plusieurs ensembles de données ré-échantillonnées à partir de l'ensemble des données observées et à l'aide d'un tirage aléatoire avec remise (voir Figure 3.6).

Supposons que l'on dispose d'un ensemble$T$ de N donner observées de notre population et que l'on s'intéresse à un statistique notée S(T). Le bootstrap va consister à former L échantillons, où chaque échantillon est constitué par tirage aléatoire avec remise de N données chaque $T * k$ est constitué par tirage aléatoire avec remise de N données dans T échantillons appelés *bootstrap*.

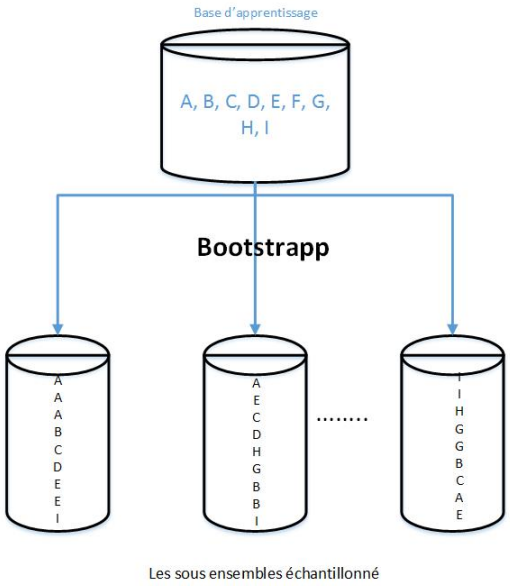

FIGURE 3.6 – Principe de Bootstrapping

L'idée majeure est donc d'obtenir divers classifieurs, basés sur de nombreux échantillons. L'algorithme peut alors s'écrire de la manière suivante (voir Algorithme 2) :

Algorithme : Bagging
; **Entrées :** $S = \{f(x_1; y_1); \cdots; (x_m; y_m)\}$, l'ensemble d'entrainement, T , le nombre d'itérations.
Pour $t = 1; ...; T$ **faire**
    $S_t \leftarrow 0$
    Pour $i = 1; ...; m$ **faire**
        tirer un exemple au hasard avec remise dans S et l'ajouter dans S_t.
    FinPour
    $h_t = A(S_t)$
FinPour

Sorties : $H(x) = sign(\sum_{t=1}^{T} h_t(x))$

Algorithme 2 : pseudo code de l'approche Bagging

D'après Breiman [BB96], la statistique $S(T)$ que l'on cherche à étudier est un algorithme d'apprentissage noté $H(x)$. Il a appliqué alors le principe de bootstrap. Ainsi chaque classifieur élémentaire $h(x)$ de l'ensemble sera entrainé sur un des L échantillons bootstrap de sorte qu'ils soient tous entraînés sur un ensemble d'apprentissages différent. Ce paramètre introduit par la méthode bootstrap permet l'évaluation interne du classifieur et l'estimation de l'importance des variables pour la sélection de variables.

La principale force du Bagging est de réduire l'instabilité d'un classifieur. De ce fait, Breiman a fait le choix d'application de classifieurs de type arbres de décision, parce qu'un petit changement dans la base d'apprentissage provoque un changement important dans la structure de l'arbre et donc dans ses performances en généralisation. Réduire l'instabilité permet alors dans ce cas de fiabiliser les prédictions et d'améliorer les performances en généralisation. Or, les arbres de décision sont des classifieurs très instables [GEN10].

Il existe un autre point qui fait la force du bagging, ce sont les mesures out-of-bag. *La mesure des Out-Of-Bag* est un paramètre qui permet d'ajouter les instances non tiré par l'échantillonnage aléatoire avec remise. Ce paramètre introduit par la méthode bootstrap permet l'évaluation interne du classifeur et l'estimation de l'importance des variables pour la sélection de variables [Bre01].

3.2 Méthodes d'ensemble séquentielles

Les méthodes d'ensembles séquentiels fonctionnent d'une manière itérative et récurrente, le boosting est parmi les méthodes d'ensembles séquentielles les plus abouti.

Boosting

Le Boosting est une méthode d'ensemble introduite par Schapire [FS99], son but est d'améliorer la performance d'un algorithme d'apprentissage. Théoriquement, le

Boosting peut améliorer significativement les performances de tout algorithme d'apprentissage caractérisé comme étant faible, c'est-à-dire, un algorithme qui n'est assuré que de retourner un classifieur de risque élevé, mais inférieur à 50% [ZIR13].

Pour illustrer le problème du boosting, nous reprenons l'exemple du parieur de courses hippiques, cherchant à maximiser ses gains, il décide de créer un programme informatique prédisant le vainqueur d'une course de chevaux. Pour concevoir ce programme, il fait appel à un expert, et lui demande d'expliquer sa stratégie de jeu. Bien évidemment, l'expert a toutes les peines du monde à exprimer "littéralement" des règles automatiques visant à miser sur tel ou tel cheval.

Cependant, quand on lui présente un ensemble de courses passées avec leurs résultats, il est tout à fait capable d'extraire quelques règles, du type : "mise sur ce cheval qui a gagné le plus de courses récemment". Si on lui présente une autre série de courses, il sera capable de compléter son expertise par d'autres règles, par exemple, "ce cheval est plus à l'aise sur courtes distances", etc.

Bien que chaque règle soit individuellement peu performante, on peut tout au moins espérer qu'elle obtienne des résultats meilleurs qu'un tirage aléatoire. Afin d'obtenir un programme performant, notre parieur doit donc résoudre deux problèmes :

1. Comment doit-il choisir les séries de courses présentées à l'expert, afin que les règles générées soient les plus utiles ?
2. Une fois collectées, les différentes règles, individuellement peu performantes, comment combiner celles-ci en une règle de décision unique et efficace ?

Pour répondre à la première question le boosting génère un ensemble diversifié d'hypothèses (classifieurs) par la modification de la distribution de l'échantillon en faveur des exemples difficiles à apprendre, c'est-à-dire, les exemples d'apprentissage sur lesquels les hypothèses sont mal classé. Et pour la seconde question, le boosting combine les hypothèses à l'aide d'un vote, où chaque hypothèse se voit pondérée en fonction de sa performance [Suc06].

Le Boosting s'appuie sur le même principe que le Bagging (voir Figure 3.7) : il construit un ensemble de classifieurs qui sont ensuite agrégés par une moyenne pondérée des résultats. Cependant, dans le cas du Boosting, cet ensemble de classifieurs est construit d'une façon récurrente et itérative. plus précisément, chaque classifieur est une version adaptative du précédent en donnant plus de poids aux observations mal prédites [CHA14].

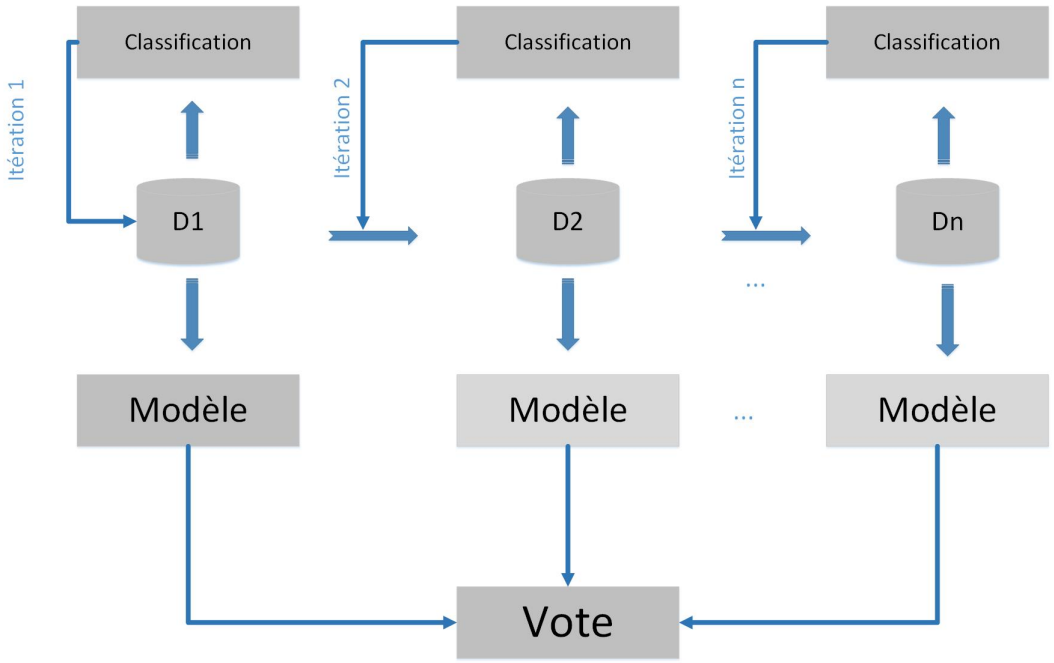

FIGURE 3.7 – Principe de Boosting

Une illustration claire du Boosting est donnée par Adaboost [FS99], un algorithme d'usage très courant en apprentissage automatique, dû à sa simplicité et à ses performances et dont le pseudo-code est détaillé par l'algorithme ci-dessous. L'expression d'Adaboost est une concaténation de "Adaptative Boosting". Il construit un vote de majorité itérativement de manière adaptative. À la fin de l'algorithme, chaque classifieur h_t est pondéré par une valeur α_t calculée dans la boucle de l'algorithme. La classification d'un nouvel exemple se fait en utilisant un vote de majorité pondéré [Suc06].

1 **Algorithme :** Adaboost

Entrées : Une base d'exemple $(x_1, y_1), ..., (x_m, y_m)$, Avec des labels $y_i \in \{1, 1\}$

2 **Pour** *t = 1, ...,T* **faire**

- On trouve le classifieur faible minimisant un critère d'erreur;
- On calcule son poids
- On calcule une distribution Dt (i) des exemples;

3 **FinPour**

Sorties : On construit le classifieur fort en effectuant une somme pondérée des classifieurs sélectionnés

Algorithme 3 : pseudo code de l'approche Adaboost

3.3 Types d'agrégation de classifieurs

Il existe trois principales catégories de vote [BAH10] :

- Vote simple (vote majoritaire) : consiste à affecter un poids équivalent à chaque classifieur et revient donc à construire un méta-classifieur correspondant à pré-

dire la classe la plus fréquemment votée par les T modèles [BK99].

– Vote pondéré : chaque classifieurs reçoit un poids α_t proportionnel à sa performance estimée sur un échantillon test. ce système de vote donne une meilleure performance par rapport au vote simple.

– Majorité pondéré : l'algorithme de majorité pondéré [LW94], il est semblable au vote pondéré, à la différence que les poids α_t attribués aux différents classifieurs formant le comité sont appris aux-même par un algorithme d'apprentissage.

3.4 Comparaison entre les méthodes d'ensemble

L'étude comparative de Dietterich [Die00], montre que dans le cas des données déséquilibré, Adaboost est plus performant que le bagging et qu'il est moins risqué au cas du sur-apprentissage, car Adaboost essaye directement d'optimiser les votes pondérés. Cette théorie est prouvée par :

1. La diminution exponentielle de l'erreur empirique d'adaboost sur l'échantillon d'apprentissage avec le nombre d'itération,
2. la diminution de l'erreur en généralisation qui continue à baisser même lorsque l'erreur empirique à atteint son minimum.

Contrairement au Bagging qui demande un grand nombre d'itération pour que l'erreur en généralisation se stabilise, l'erreur de généralisation du Boosting ne diminue pas lorsque l'erreur en apprentissage a atteint son minimum.

C'est ainsi que le Bagging est plus gourmand que le Boosting du point de vue d'espace mémoire puisque chaque classifieurs doit être stocké pour la prédiction d'un nouvel exemple [BAH10].

Bagging	**Boosting**
Aléatoire	Adaptatif et généralement déterministe
Utilise des échantillons Bootstrap	Utilise échantillon initial au complet
Les modèles ont le même poids	Les modèles pondérés selon leur qualité d'ajustement

TABLE 3.1 – Bagging vs Boosting

3.5 Le Boosting face aux données déséquilibrées

L'algorithme principal du boosting est l'adaboost [FS99], c'est une procédure adaptative de construction d'un ensemble de classifieurs, sont principe est de créer à la première itération un classifieurs faible ensuite les instance mal classés seront sur-pondérés. Les diverses pondérations permettent de construire un nouveau échantillon d'apprentissage. Finalement, un vote pondéré de l'ensemble des classifieurs est appliqué pour obtenir la décision finale.

Il est raisonnable de penser que grâce à cette mise à jour adaptative, le boosting tiens compte du déséquilibre de classes et améliore la performance de la classification face à ce type de données. En effet, les premières itérations auront tendance à bien classer les exemples issus de la classe majoritaire, toutefois au fur est à mesure des itérations, les exemples de classe minoritaire seront de plus en plus on surpondérés, jusqu'à devenir les exemples à bien classer en priorité par le classifieurs courant [BAH10].

4 Les méthodes hybrides

Due aux performances de méthodes d'ensembles et les méthodes d'échantillonnages aux domaine de déséquilibre de classes, plusieurs chercheurs ont pensés de combiner les approches pour plus de performance.

4.1 Échantillonnage combiné avec Boosting

Dans la littérature, plusieurs algorithmes de type de boosting spécifiquement dédiés au problème du déséquilibre des classes ont était proposé, on citera d'abord adacost [FSZC] qui utilise une matrice de coût de mauvaise classification pour repondéré les exemples, RareBoost [JKA01] re-pondère les exemples en fonction du rappel et de la précision.

Nous trouvons également SMOTEBoost [SKHN10] celle qu'on a choisis pour notre étude, SMOTEBoost consiste à utiliser l'algorithme SMOTE avant d'appliquer AdaBoost sur le nouvelle échantillon construit par SMOTE.
Le SMOTE est utilisé pour l'amélioration de la précision de la classe minoritaire et le Boosting pour l'amélioration de la précision de la base entière. Le but principal de SMOTEBoost est de mieux modéliser la classe minoritaire dans la base entière.

Le boosting donne des poids égaux pour toute les instances mal classé même les instances de la classe majoritaire, le boosting traite de manière similaire les 2 erreurs (FP, FN).

L'objectif est de réduire le biais inhérent à l'apprentissage en raison de déséquilibre de classe et augmenter le poids d'échantillonnage de la classe minoritaire.

Une autre technique de combinaison le sous-échantillonnage aléatoire avec l'Adaboost (RUSBoost) fonctionne d'une façon similaire à SMOTEBoost, mais il supprime des instances de la classe majoritaire par le sous-échantillonnage aléatoire à chaque itération de l'apprentissage. [YZZJ14].

4.2 Échantillonnage combiné avec Bagging

Plusieurs approches sont développées utilisant le bagging pour traiter la classe déséquilibré. Reconnue pour sa simplicité, sa capacité de généralisation, l'hybridation de la méthode bagging avec une technique de pré-traitement de données est plus simple par rapport à son intégration au Boosting, le bagging n'a pas besoin

d'attribuer des poids donc il ne nécessite pas d'adapter la formule de mi à jour de poids ni de modifier les calculs de l'algorithme.

Dans cette méthode, la manière de fonctionnement est basée sur la façon de générer les sous-ensembles bootstrap et comment la classe déséquilibré est représenté pour avoir un classifieur performant à chaque itération, sans oublier l'importance de diversité de classe, on distingue 2 types d'algorithmes : SMOTEBagging [YMD+13], UnderBagging [WY09].

SMOTEBagging est la combinaison de la méthode de sur-échantillonnage SMOTE avec la méthode d'ensemble Bagging, la façon de son fonctionnement est à chaque itération, on effectue un taux d'échantillonnage (allant de 10% pour la première itération jusqu'à 100% pour la dernière itération) ce taux définie le nombre des instances positifs échantillonné aléatoirement et avec remise par le bootstrap.

La deuxième technique est UnderBagging c'est la combinaison de sous- échantillonnage aléatoire avec le Bagging, c'est une méthode très simple son principe de fonctionnement est à chaque sous ensemble on effectue un sous-échantillonnage aléatoire avant la formation des classifieurs pour chaque itération de bagging sous-échantillonnage élimine les instances majoritaire.

5 Conclusion

Dans ce chapitre, nous avons présenté tous les approches utilisées pour surmonter le problème de déséquilibre de données qui sont les méthodes d'échantillonnages, les méthodes d'ensembles, et la combinaison entre les deux.

Par la suite de ce travail, nous allons étudier l'application de ces méthodes avec différentes manières et découvrir quelle est la meilleure méthode qui donne la plus grande précision.

Chapitre 4

Expérimentations et Résultats

1 Introduction

Nous présentons dans ce chapitre, notre étude expérimentale répartie en 3 parties appliquant les méthodes les plus récentes utilisées pour surmonter le problème de déséquilibre de classes sous deux environnements différents (Java-Keel [AFSG+09] et logiciel libre R [IG96]). La démarche adaptée pour notre travail est résumé dans les points suivant (voir Figure 4.1) :

- Expérimentation 1 : Étude comparative entre différentes méthodes d'ensembles sur 3 bases de données sous Keel.
- Expérimentation 2 : Analyse de l'influence de SMOTE sur différents degrés de déséquilibre sous Keel.
- Expérimentation 3 : Application du sur et sous échantillonnage sur l'algorithme Adaboost sous R.

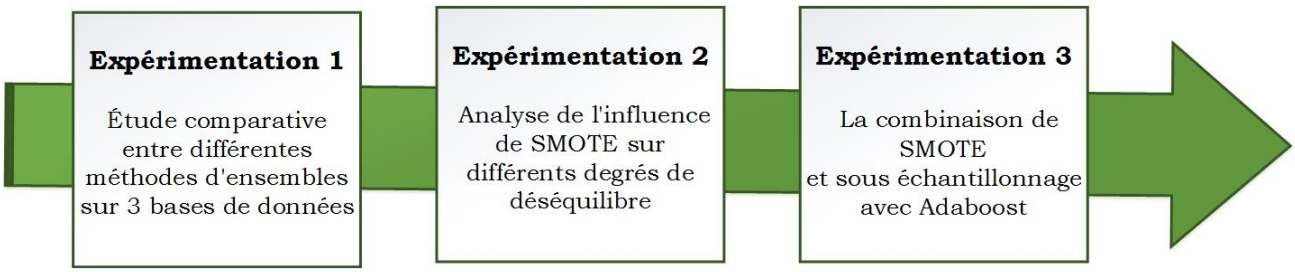

FIGURE 4.1 – Processus de l'étude expérimentale

2 Bases de données

Pour la réalisation de notre travail, nous appliquons 6 bases de données médicales extraites du dépôt d'UCI [Lic13] : Haberman, wisconsin, liver-disorder, EEGeye, BreastTissue.

2.1 Haberman

L'ensemble de données contient des cas d'une étude qui a été accomplie entre 1958 et 1970 à l'Université Chicago's Billings Hospital sur la survivance de patients qui avaient subi la chirurgie pour le cancer du sein . La base de données contient 306 instances (225 survécu et 81 mort). Elle contient trois attributs et un quatrième pour la classe chiffré par 1 si le patient est mort pendant 5ans et 2 s'il est survécu 5ans ou plus [ALL13].

Les descripteurs cliniques de la base Haberman :

1. Age : L'âge de patient au temps d'opération (numériques).
2. Year : L'année d'opération du patient (l'année - 1900, numérique)
3. Positive : Le nombre de noeuds axillary positifs a découvert (numériques).
4. Variable classe : Le statut de survivance (1 ou 2)

2.2 Liver-disorder

La base de données Liver disorders a été dénotée par Richard S. Forsyth dans une recherche médicale de la compagnie de soins médicaux internationaux BUPA.
Elle réalise une étude médicale sur 345 individus des maladies du déséquilibre de foie (200 malades, 145 non malades).
La base de données contient six attributs, les cinq premiers sont toutes les analyses de sang qui sont pensés être sensibles aux désordres de foie qui pourraient émaner de la consommation d'alcool excessive.

Chaque ligne dans l'ensemble de données est constituée d'un enregistrement d'un seul individu mâle.

Il semble que les boissons$>$ 5 sont une sorte d'un sélectionneur sur cette base de données [ALL13].

Les descripteurs cliniques de la base :

1. MCV : Le volume corpusculaire.
2. alkphos : Phosphotase alcalin
3. sgpt : Alamine aminotransferase.
4. sgot : Aspartate aminotransferase
5. gammagt : Gamma-glutamyl transpeptidase
6. drinks : Le nombre d'équivalents demi-d'une pinte de boissons alcoolisées bues par jour
7. Classe : Variable de classe (1 ou 2)

2.3 Breast Cancer (Wisconsin)

La base de données du cancer du sein dénommée Wisconsin Breast Cancer Database a été collectée à l'Université du Wisconsin. Elle contient les informa-

tions médicales de 699 cas cliniques relatifs au cancer du sein classés comme bénin ou malin : 458 patientes (soit 65%) sont des cas bénins et 241 patientes (soit 35%) sont des cas malins.
La base de données contient neuf attributs qui représentent des cas cliniques, et l'attribut de la classe avec un diagnostic chiffré par 2 si le cas est bénin ,4 si le cas est malin [ALL13].

Les descripteurs cliniques de la base Breast Cancer :

1. ClumpThickness : l'épaisseur de la membrane plasmique d'une cellule cancéreuse.
2. CellSize : L'uniformité de la taille d'une cellule cancéreuse.
3. CellShape : L'uniformité de la forme d'une cellule cancéreuse.
4. MarginalAdhesion : Adhesion marginale (une surexpression de la protéine integrin beta3 au niveau de la surface de la cellule cancéreuse).
5. EpithelialSize : Taille cellule épithéliale (La détection des cellules épithéliales dans la moelle osseuse).
6. BareNuclei : Bare Nuclei (La détection les nucléoles qui se trouvent à l'exterieur du noyau).
7. BlandChromatin : Bland Chromatin (une protéine qui induit l'expression du gène du récepteur d'oestrogènes).
8. NormalNucleoli : Normal Nucleoli (L'ADN protégé par une membrane nucléaire)..
9. Mitoses : Mitoses (La mitose est un processus de division cellulaire régulé).
10. Variable de classe (1 ou 2).

2.4 EEG eye

la bases est construit à partir d'un EEG continue, mesuré avec Emotiv EEG Neuohead set pendant 117 secondes aux différents états de l'œil (ouvert, fermé) détecte a laide d'une camera. La base contient 14977 instances avec 15 attributs qui représentent les valeurs des électrodes et l'état de l'œil. 8255 (soit 55,12%) instances correspondes aux examens avec l'œil ouvert, et 6722 (soit 44,88%) instances correspondes à l'œil fermé [RS13].

2.5 Breast Tissue

représente les informations sur des mesures d'impédances électriques dans des échantillons de tissus de sein [Lic13].

Cette base contient 106 instances et 9 attributs :

1. I0 : mesure d'impédance à la fréquence 0
2. PA500 : angle de phase à 500 KHz
3. HFS : haute fréquence d'angle de phase

4. DA : la distance entre les extrémités d'impédance spectrales
5. AREA : surface sous spectre
6. A/DA : zone normalizer par DA
7. A/DA : zone normalizer par DA
8. MAX IP : maximum du spectre
9. DR : distance entre I0 et partie réelle du point de fréquence maximale
10. P : longueur de la courbe spectrale.

les classe de tissue sont représenté comme suit :

Tissue normal :

1. connective tissue(con) 14 cas
2. adipose tissue (adi) 22 cas
3. glandular tissue (gla) 16 cas

Tissue pathologiquel :

1. Carcinoma (car) 21 cas
2. Fibro-adenoma (fad) 15 cas
3. Mastopathy (mas) 18 cas [EdSMdSJ00]

2.6 Heart

La base heart diseases représente les maladie de cœur , 303 instances et 76 attribut Mais la majorité des expériences utilisent la version qui contient 14 attributs.
La base est regroupé en 5 classes, la classe 0 représente l'absence de la maladie, et 1,2,3,4 la présence d'une maladie. [Lic13]

3 Matériels et méthodes

Les méthodes ensemblistes ont permis d'améliorer de façon spectaculaire des classifieurs standard en apprentissage supervisé et en particulier lors de présence de données en distribution déséquilibrée. De ce fait, dans notre étude expérimentale, nous avons testé sur des données réelles médicales les différentes méthodes d'ensemble qui sont présentées dans l'organigramme suivant :

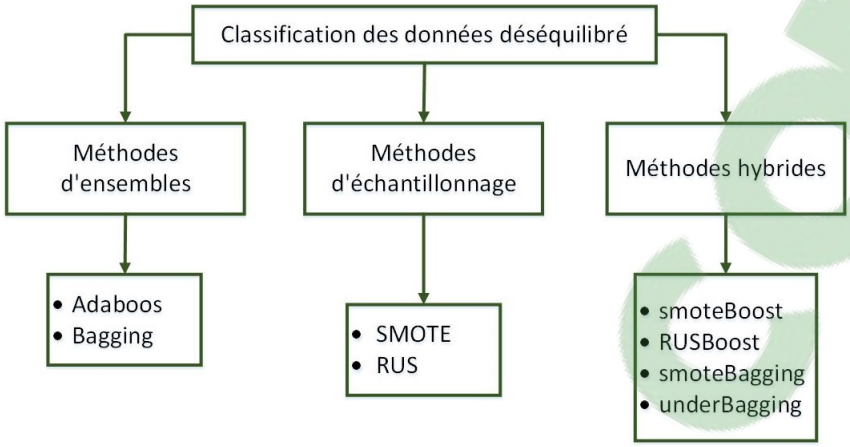

FIGURE 4.2 – Algorithmes utilisés

Adaboost [FS99] : c'est une méthode d'ensemble séquentiel la plus utilisée,son principe est de lancer l'apprentissage sur toute la base et à chaque itération d'apprentissage il donne l'importance aux instances mal classé.

Bagging [BB96] : c'est une méthode d'ensemble parallèle son principe est de ré-échantillonné la base d'apprentissage en sous ensembles aléatoirement avec remise (Bootstrapping) et pour chaque sous ensemble on affecte un classifieur.

SMOTE(Synthetic Minority Over-Sampling TEchnique) : [CBHK02] est méthode de sur-échantillonnage permet de générer des instances minoritaire synthétiquement

Sous-échantillonnage aléatoire : [BPM04] permet de supprimer les instances majoritaire aléatoirement.

SMOTEBoost : [SKHN10] combaision de la méthodes de prétraitement SMOTE avec méthodes d'apprentissage adaboost

SMOTEBagging [YMD+13] : combination de la méthodes de prétraitement SMOTE avec méthodes d'apprentissage Bagging

RUSBoost [YZZJ14] : Le sous- échantillonnahe aléatoire+adaboost

UnderBagging [WY09] : le sous-échantillonnage aléatoire+ Bagging

4 Mesures de performance

La mesure de performances est une factrice clé à la fois dans l'évaluation de la performance de classification et d'orientation de la modélisation de classifieur. Pour comparer de façon synthétique, différents méthodes de mesure de performance sont retenues pour nos expériences : Accuracy, Sensibilité, Spécificité,

G-means [Hao14] :

$$Acc = \frac{VP + VN}{VP + VN + FP + FN}$$

$$Sen = \frac{VP}{VP + FN}$$

$$Spe = \frac{VN}{VN + FP}$$

$$G - means = \sqrt{Sensibilit \times Spcificit}$$

- Vrai positive(VP) : les cas positives classé positives
- Vrai negative(VN) :les cas négative classé negative
- Faux positive(FP) : les cas positive classé négative
- Faux négative(FN) : les négative classé positive
- Accuracy(Acc) : le pourcentage des exemples classé correctement
- Sensibilité(Sen) : le pourcentage des exemples positive classé correctement
- Spécificité(Spe) : le pourcentage des instances négative classé correctement
- G-means : Il fournit un moyen simple d'évaluer la capacité du modèle à classer correctement, par la combinaison de sensibilité et spécificité dans une seule métrique. G-means est considérée comme une mesure de la précision équilibrée

5 Expérimentation 1 : Étude comparative entre différentes méthodes d'ensembles

Cette partie relate une étude comparative entre les méthodes les plus récentes utilisées pour la classification de données déséquilibrées : méthodes d'échantillonnage, méthodes d'ensembles ainsi que les méthodes hybrides.

La question que nous nous posons est : *quelle est la meilleure méthodes ?*

Les expérimentations ont été réalisées sur 3 bases de données médicales binaire : Liver-disorder, Wisconsin, Haberman issues du dépôt de données UCI Machine Learning [Lic13]. Le tableau (Table 4.1) résume les caractéristiques de chaque base.

Bases	Entrés	Taille de la base	Base app	Base test	Degré de déséquilibre
Wisconsin	9	679	546	133	34,9%
Liver-disorder	6	345	259	86	42,08%
Haberman	3	306	230	76	26,08%

TABLE 4.1 – Description des bases de données

Nous avons divisé les bases originales en deux sous bases d'apprentissage et de test. 75% de la base originale pour l'apprentissage et le reste pour le test (25%).

Les algorithmes utilisés sont représentés sous forme d'organigramme (voir Figure 3).

L'évaluation de ces différentes techniques, ont été réalisées sous le logiciel Keel sofware [AFSG$^+$09], les résultats obtenues sont résumé dans le tableau 4.2 suivant :

Bases de données	**Test d'évaluation**	***Ada-Boost***	***SMOTE-Boost***	***Under-Boost***	***Bagging***	***SMOTE-bagging***	***Under-bagging***
Haberman	***Acc***	60,51%	68,78%	60,51%	57,35%	63,80%	67,53%
	Sen	42,85%	66,66%	42,85%	23,80%	47,61%	71,42%
	Spe	78,18%	70,90%	78,18%	90,90%	80,00%	63,63%
	G-mean	57,87%	68,74%	57,87%	46,51%	61,71%	67,41%
Wisconsin	***Acc***	97,35%	97,81%	95,63%	97,35%	96,08%	94,79%
	Sen	95,83%	98,33%	96,66%	95,83%	96,66%	95,00%
	Spe	98,87%	97,29%	94,59%	98,87%	95,49%	94,59%
	G-mean	97,33%	97,80%	95,61%	97,33%	96,07%	95,29%
Liver-disorder	***Acc***	67,12%	67,45%	58,60%	62,05%	65,41%	56,56%
	Sen	64,86%	67,56%	70,27%	56,75%	67,56%	70,27%
	Spe	69,38%	67,34%	46,93%	67,34%	63,26%	42,85%
	G-mean	67,08%	67,44%	57,42%	61,81%	65,37%	54,87%

TABLE 4.2 – Tableau comparatif des résultats de classification des différentes approches sur les 3 bases de données.

Interprétation des résultats

Nous abordons en premier lieu, la comparaison des performances réalisées par l'utilisation des méthodes d'ensemble (bagging, adaboost).

Les deux mesures accuracy et G-means montrent que pour les 3 bases l'adaboost est plus performant que le bagging. Cela du a son principe même qui se base sur la sélection itérative de classifieur faible en fonction d'une distribution des exemples d'apprentissage. Chaque exemple est pondéré en fonction de sa difficulté avec le classifieur courant.

La sensibilité est élevée avec l'utilisation d'adaboost, ce qui confirme sa performance sur les instances minoritaire.

La spécificité a donnée des bons résultats pour le bagging avec 2 bases (wisconsin, haberman), ce qui démontre que le bagging donne l'importance à la classe majoritaire.

En second lieu, nous comparons la combinaison des méthodes d'ensembles par les deux méthodes d'échantillonnages (sur-échantillonnage, Souséchantillonnage). Le SmoteBoost et UnderBoost sont plus performants que l'adaboost tout seul, aussi la sensibilité augmente considérablement. La remarque qui peut être faite est au niveau de la spécificité, qui diminue de manière significative, ce qui nous conduit à dire que les méthodes d'échantillonnages sont capables de classer que les données minoritaires. Les mêmes résultats sont observés pour le SmoteBagging et UnderBagging.

De manière globale, la méthode optimale qui réalise les plus hautes performances sur les 4 bases utilisées est SMOTE-Boost. Grâce à son principe qui se base sur la création d'exemples de synthèse de la classe minoritaire, ainsi indirectement les poids changent de mise à jour et de compensation pour les distributions asymétriques. SMOTEBoost appliqué à plusieurs ensembles de données hautement et modérément déséquilibrées montre amélioration de la performance de prédiction sur la classe minoritaire. De ce fait, le SMOTEBoost apporte le meilleur compromis est jugé comme meilleur dans cette première expérimentation.

6 Expérimentation 2 : Application de l'approche SMOTE sur différents degrés de déséquilibres

La deuxième partie de ce travail, consiste à réaliser le changement du degré de déséquilibre en utilisant l'algorithme adaboost avec et sans la méthode d'échantillonnage SMOTE. Nous étudions ici, l'intérêt de l'application de la méthode SMOTE et si elle apporte une amélioration à la classification sur différents degrés de déséquilibre.

La question qui est posée à ce niveau est *L'approche SMOTE est elle meilleure et robuste pour tous les niveaux de déséquilibre ?*

Dans cette expérimentation, nous avons fixé 5 degrés de déséquilibre 30%, 20%, 10%, 5%, 1%. Le tableau Table 4.3 englobe les résultats réalisés.

	teste d'évaluation	Haberman		Wisconsin		Liver-disorder	
		Adaboost	**SMOTEBoost**	**Adaboost**	**SMOTEBoost**	**Adaboost**	**SMoTEBOst**
30%	Acc(%)	62,68	67,09	96,14	97,81	65,47	73,24
	Se(%)	38,09	52,38	95,00	98,33	51,35	64,86
	Sp(%)	87,27	81,81	97,29	97,29	79,59	81,63
	G-mean(%)	57,65	65,46	96,13	97,8	63,92	72,76
20%	Acc(%)	51,12	66,75	96,14	98,26	68,56	64,17
	Se(%)	09,52	57,14	95,00	98,33	43,24	32,43
	Sp(%)	92,72	76,36	97,29	98,19	93,87	95,91
	G-neam(%)	29,71	66,05	94,64	98,27	63,7	55,77
10%	Acc(%)	47,27	55,19	92,04	94,09	54,05	64,17
	Se(%)	0	28,57	85,00	90,00	08,10	32,43
	Sp(%)	94,54	81,81	99,09	98,19	100	95,91
	G-mean(%)	0	48,34	91,77	94,00	28,46	55,77
5%	Acc(%)	54,41	61,21	86,21	94,54	54,05	57,08
	Se(%)	14,28	33,33	73,33	90,00	08,10	16,21
	Sp(%)	94,54	89,09	99,09	99,09	100	97,95
	G-mean(%)	36,74	54,49	85,24	94,43	28,46	38,84
1%	Acc(%)	50,00	52,94	50,00	93,26	51,35	54,05
	Se(%)	0	09,52	0	88,33	02,70	08,10
	Sp(%)	100	96,36	100	98,19	100%	100
	G-mean(%)	0	30,28	0	93,12	16,43	28,46

TABLE 4.3 – Tableau comparatif des résultats de classification des différentes degres de déséquilibre avec l'approche SMOTE Adaboost sur les 3 bases de données

Interprétation des résultats

Nous avons effectué cinq changements de degrés de déséquilibre sur chacune des 3 bases. L'évaluation de ces dernières a était fait par les mesures Accuracy, Sensibilité, Spécificité, G-means.

Nous remarquons que les valeurs d'Accuracy et G-means augmentent avec l'utilisation de SMOTE sur toutes les bases, cela s'explique par son principe de création de nouvelles instances minoritaires.

Lors de l'utilisation d'Adaboost sans SMOTE, l'accuracy diminue avec le changement de déséquilibre. Il atteint les 47% pour Haberman ce qui conduit à une sensibilité plus basse de 0% dans les deux degrés de déséquilibre 10% et 1%. Pour la base liver-disorder un pourcentage de 2,7% au niveau de 1% de déséquilibre. Ceci montre une faible reconnaissance des instances minoritaires. Par contre, la spécificité augmente lors de la diminution de degré de déséquilibre où la classe majoritaire devient de plus en plus dominante.

La combinaison de SMOTE avec Adaboost montre certaine amélioration par rapport à l'utilisation de d'Adaboost seul au niveau des 5 degrés de déséquilibre. La sensibilité est nettement améliorée dans les différentes bases de données, ce qui confirme la performance de SMOTE sur le problème de déséquilibre, mais la spécificité est relativement diminué, ce qui peut être traduit par le fait que l'algorithme SMOTE donne la priorité aux données minoritaires.

7 Expérimentation 3 : La combinaison de SMOTE et sous échantillonnage avec Adaboost

.
Cette partie présente l'utilisation de SMOTE avec le sous échantillonnage aléatoire et l'adaboost (voir Figure 4.3). Nous avons appliqué cette technique sous l'environnement R en utilisant les 2 package DMwR [Tor10] (pour l'échantillonnage) et Ada [CJM06] (pour adaboost). Le SMOTE sous R est considéré comme une technique de sur et sous échantillonnage au même temps, il ajoute des instances minoritaires et supprime les instances majoritaires.

Le principe de SMOTE sous R, commence par fixer les pourcentages de sur-échantillonnage et sous-échantillonnage. Le taux de sous-échantillonnage est indépendant de nombre d'instances majoritaires, il dépend des instances minoritaires après le sur-échantillonnage par SMOTE.

Par exemple, dans notre application, nous avons sous-échantillonné la classe majoritaire à 200%, cela signifie que l'ensemble de données modifié contiendra 2 fois autant d'éléments de la classe minoritaire à partir de la classe majoritaire (la classe minoritaire, 2 fois, la classe majoritaire) [CBHK02].

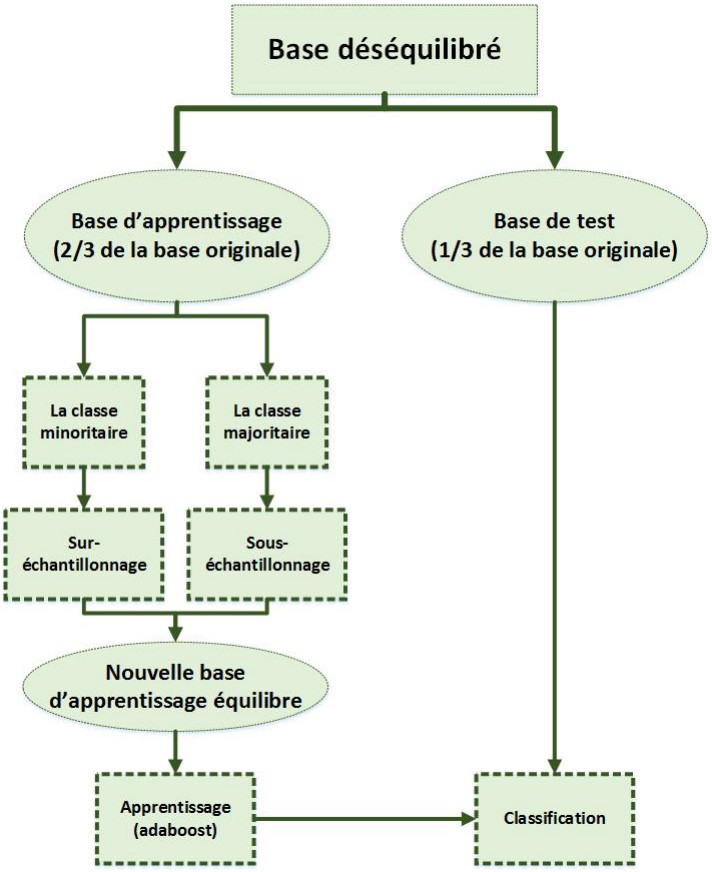

FIGURE 4.3 – Principe de SMOTE sous R

Dans cette expérimentation, nous avons ajouté 3 bases de données volumineuses EEG eye, et 2 multi classes Heart et BreastTissue (Table 4.4).

Bases de données	**Entrées**	**Taille de la base**	**Base app.**	**Base test**	**% déséquilibre**
EEG	15	14977	9987	4990	29,6%
BreastTissue	9	106	71	35	12,67%
Heart	14	115	76	35	6,65

TABLE 4.4 – la description des bases de données

Nous avons varié le nombre le nombre d'instances minoritaires et majoritaires, parfois le nombre de classes majoritaires est supérieur au nombre de classes minoritaires et parfois le contraire, jusqu'à l'obtention de l'équilibre totale. (nombre d'instances majoritaire = nombre d'instances minoritaire).

Nous avons fixé le taux de sur-échantillonnage à 100% et le taux de sous-échantillonnage à 50%, 100%, 150%, 200%, 300%, 400%.

BDD	Mesure d'évaluation	50%	100%	150%	200%	300%	400%
Haberman	Sensibilité	88,88%	77,77%	74,07%	74,07%	51,85%	55,55%
	Spécificité	28%	44%	54,66%	60%	80%	85,33%
	Accuracy	44,1%	59,9%	59,8%	63,7%	72,5%	77,5%
	G-means	49,88%	58,49%	63,62%	66,66%	64,40%	68,84%
Liver-disorder	Sensibilité	89,83%	72,88%	66,1%	52,54%	47,45%	47,45%
	Spécificité	26,78%	51,78%	60,71%	69,64%	73,21%	73,21%
	Accuracy	59,1%	62,6%	63,5%	60,9%	60%	60%
	G-means	49,04%	61,43%	63,34%	60,48%	58,93%	58,93%
Wisconsin	Sensibilité	96,15%	94,87%	93,58%	93,58%	93,58%	93,58%
	Spécificité	96,64%	97,31%	97,98%	97,98%	97,31%	97,98%
	Accuracy	96,5%	96,5%	96,5%	96,5%	96%	96,5%
	G-means	96,39%	96,08%	95,75%	95,78%	95,42%	95,75%
Breast Tissue	Sensibilité	98,51%	96,11%	94,80%	94,27%	92,09%	91,48%
	Spécificité	72,30%	86,42%	91,05%	93,52%	95,11%	96,07%
	Accuracy	84,3%	90,9%	92,8%	93,9%	93,7%	94%
	G-means	84,39%	91,13%	92,90%	93,89%	93,58%	93,74%
heart	Sensibilité	100%	100%	100%	50%	0%	0%
	Spécificité	0%	0%	0%	53,84%	100%	100%
	Accuracy	4,87	4,87	4,87	53,65%	95,12%	95,12%
	G-means	0%	0%	0%	58,88%	0%	0%
Eeg Eye	Sensibilité	98,29%	96,41%	95,01%	93,35%	92,44%	90,95%
	Spécificité	65,08%	87,75%	91,49%	93,67%	95,45%	96,22%
	Accuracy	84,31%	91,72%	68,18%	93,53%	94,07%	93,81%
	G-means	79,97%	91,97%	93,23%	93,65%	93,93%	93,54%

TABLE 4.5 – Resultats de la combinaison de SMOTE avec sous-échantillonnage aléatoire

Interprétation des résultats

Au niveau de 50%, 100%, 150%, le nombre d'instances minoritaires est supérieur au nombre d'instances majoritaire. La sensibilité est supérieure à la spécificité le classifieur reconnue la classe la plus nombreuse.

Contrairement au niveau de 300% et 400% où la classe majoritaire est plus grande que la classe minoritaire, chaque classe peu nombreuse est traité comme bruit, c'est l'exemple de la base Heart où nous avons obtenu 0% de sensibilité sur 50%, 100%, 150% et 0% de spécificité sur 300%, 400%.

Au niveau de 200%, la taille des 2 classes sont égales, la sensibilité et la spécificité sont très proches et le taux de reconnaissance est le même pour les 2 classes due à l'équilibrage de la distribution de classes.

8 Conclusion

Les données déséquilibrées sont un problème très souvent rencontré dans le domaine d'aide au diagnostic. Il peut causer des effets négatifs sérieux sur les performances de classification de l'apprentissage.

Ce chapitre, présente l'application de différentes approches existantes dans la littérature dans le but de surmonter le problème de déséquilibre de données.

Deux grandes familles de technique sont appliquées, les méthodes d'ensemble et méthodes d'échantillonnage. Des expérimentations réalisées, nous pouvons conclure que la méthode de sur-échantillonnage SMOTE a bien amélioré la distribution de classes par la création de nouvelles instances minoritaires, ainsi que la méthode d'ensemble Adaboost qui a été bien adaptée au déséquilibre par son principe qui corrige les erreurs de classification itérativement, elle a donné des meilleures performances pour les deux classes (minoritaire, majoritaire).

De ce fait, leur hybridation est plus performante, car d'une part, il équilibre la base par la méthode SMOTE d'autre par apporter une meilleure précision par le principe de l'algorithme d'adaboost.

Conclusion générale

Durant ce projet de fin d'études, nous avons traité un problème fréquent rencontré par la nature des données médicales qui est le déséquilibre de classes. Ce problème survient lorsque l'effectif d'une classe est peu représenté par rapport aux effectifs d'autres classes.

Nous avons proposé d'apporter des solutions à ce problème dans le cadre des méthodes d'ensemble, connues pour leur pouvoir prédictif. Nous avons réalisé une étude comparative entre les différentes méthodes d'ensemble et leur hybridation par les 2 types de méthodes d'échantillonnages sur diverses bases de données.

Les résultats obtenus montrent que l'hybridation de la méthode de sur-échantillonnage SMOTE avec Adaboost est la plus performante. Par la suite, nous avons testé Adaboost avec et sans l'utilisation de SMOTE sur différents degrés du déséquilibre afin de découvrir l'influence de SMOTE sur les données déséquilibré, la conclusion faite, est que SMOTE montre une meilleure adaptation pour le déséquilibre. Finalement, pour plus de précision, nous avons ajouté à la méthode SMOTE adaboost le sous-échantillonnage aléatoire pour parvenir à un équilibre adéquat.

À travers cette étude, nous avons expérimenté les approches les plus récentes afin d'observer les effets de chacune et découvrir quelle est la méthode la plus adaptée.

Dans les perspectives d'avenir, nous prévoyons d'assurer l'interprétabilité des résultats du modèle en intégrant la notion de la logique floue, associer aux méthodes d'ensemble les techniques d'optimisation d'erreurs, et tester d'autre approches d'échantillonnages pour éviter la notion d'aléatoire.

Annexe

Keel (Knowledge Extraction based en Evolutionary Learning

La fouille de données est un processus de la découverte automatique par l'obtention de l'information du monde réel, et des bases de données larges et complexes. Elle est le noyau d'un processus appelé ECD (extraction de Connaissance à partir de Données).

Ces dernières années, plusieurs outils de fouille de données sont développés la majorité sont commercialisé et les open sources sont rares.

Keel software est un logiciel libre programmé sous java permet à l'utilisateur d'évaluer les techniques évolutives et d'autres problèmes de fouille de données Régression, Classification... etc.

Cet outil a plusieurs avantages :

* Réduis le travail du programmeur et gagner le temps .
* Une vaste bibliothèque de méthodes et bases de données prêt à tester et facile à utilisées .
* On peut l'utiliser au n'importe quelle machine avec java .

Ce logiciel est divisé en 4 grandes parties sont représenté comme suit :

– Data Management : permet la préparation de la base de données et d'autre tâche intéressante comme : la création des bases de données, créer de partition pour les bases de données existantes, en plus il est possible de modifier les bases de données et les convertir à d'autre format Excel, CSV, Weka.

– Experiments : le but de cette partie est d'utiliser les bases de données et les méthodes disponibles pour générer un répertoire avec tous les fichiers nécessaire pour l'exécution des expériences, la conception d'une expérience avec Keel est facile, il suffit de sélectionner les données d'entrées, les algorithmes, les testent d'évaluation et de faire la connexion entre eux, nous pouvons utiliser ce logiciel comme un outil de test et d'évaluation lors de l'élaboration d'un algorithme d'autre part, c'est une bonne option pour comparer les différentes méthodes

– Educational : cette partie consiste à l'exécution en ligne des expériences étant possible de s'arrêter et les reprendre comme nous avons besoin, cette partie à un nombre réduit de méthodes disponible.

– Modules : cette partie inclut de nouveaux modules étendant les fonctionnalités de l'outil logiciel de Keel :

  – Un module spécialement conçu pour générer des expériences avec des données déséquilibrées.
  – Ce module permet d'analyser facilement les résultats de toute étude expérimentale, exprimer en format CSV. Pour ce faire, il comprend plusieurs tests statistiques non-paramétriques bien connus, prêts à l'emploi.

FIGURE 4.4 – Interface de Keel software2.0

Voici la structure de Keel contient une interface bien organisée, facile à manipuler, il contient des méthodes récentes pour divers problèmes de data mining.

Logiciel R

Le logiciel R est un logiciel de statistique créé par Ihaka & Gentleman [IG96]. Il est à la fois un langage informatique et un environnement de travail : les commandes sont exécutées grâce à des instructions codées dans un langage relativement simple, les résultats sont affichés sous forme de texte et les graphiques sont visualisés directement dans une fenêtre qui leur est propre.

C'est un clone du logiciel S-plus qui est fondé sur le langage de programmation orienté objet S, développé par AT&T Bell Laboratories en 1988 [JN84]. Ce logiciel sert à manipuler des données, à tracer des graphiques et à faire des

analyses statistiques sur ces données.

R est un logiciel gratuit et à code source ouvert (opensource). Il fonctionne sous UNIX (et Linux), Windows et Macintosh. C'est donc un logiciel multi-plates-formes. Il est développé dans la mouvance des logiciels libres par une communauté sans cesse plus vaste de bénévoles motivés.

Tout le monde peut d'ailleurs contribuer à son amélioration en y intégrant de nouvelles fonctionnalités ou méthodes d'analyse non encore implémentées. Cela en fait donc un logiciel en rapide et constante évolution.

C'est aussi un outil très puissant et très complet, particulièrement bien adapté pour la mise en œuvre informatique de méthodes statistiques. Il est plus difficile accès que certains autres logiciels du marché (comme SPSS ou Minitab par exemple), car il n'est pas conçu pour être utilisé à l'aide de «clics» de souris dans des menus. L'avantage en est toutefois double :

- L'approche est pédagogique puisqu'il faut maîtriser les méthodes statistiques pour parvenir à les mettre en œuvre
- L'outil est très efficace lorsque l'on domine le langage R puisque l'on devient alors capable de créer ses propres outils, ce qui permet ainsi d'opérer des analyses très sophistiquées sur les données.

Le logiciel R est particulièrement performant pour la manipulation de données, le calcul et l'affichage de graphiques. Il possède, entre autres choses :

- Un système de documentation intégré très bien conçu (en anglais)
- Des procédures efficaces de traitement des données et des capacités de stockage de ces données
- Une suite d'opérateurs pour des calculs sur des tableaux et en particulier sur des matrices
- Une vaste et cohérente collection de procédures statistiques pour l'analyse de données
- Des capacités graphiques évoluées
- Un langage de programmation simple et efficace intégrant les conditions, les boucles, la récursivité, et des possibilités d'entrée-sortie [dMDL11].

Bibliographie

[AFSG+09] Jesus Alcala-Fdez, Luciano Sanchez, Salvador Garcia, Maria Jose del Jesus, Sebastian Ventura, JM Garrell, Jose Otero, Cristobal Romero, Jaume Bacardit, Victor M Rivas, et al. Keel : a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3) :307–318, 2009.

[ALL13] Melle Asma HAMMYANI Melle Soumia ALLIOUA. Amélioration des forêts aléatoires : Application au diagnostic médical. Master's thesis, Université Abou Bekr Belkaid, Faculté de Science, 2013.

[BAH10] Emna BAHRI. *Amélioration des méthodes adaptatives pour l'apprentissage supervisé des données réelles*. PhD thesis, Université lyon lumière 2, Faculté des Sciences Economiques et de Gestion, 2010.

[BB96] Leo Breiman and Leo Breiman. Bagging predictors. In *Machine Learning*, pages 123–140, 1996.

[Ber09] Simon Bernard. *Forêts Aléatoires : De l'Analyse des Mécanismes de Fonctionnement à la Construction Dynamique*. PhD thesis, Université de Rouen, 2009.

[BK99] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. *Machine learning*, 36(1-2) :105–139, 1999.

[BM12] Iain Brown and Christophe Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3) :3446–3453, 2012.

[BPM04] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1) :20–29, 2004.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote : Synthetic minority oversampling technique. *J. Artif. Int. Res.*, 16(1) :321–357, June 2002.

[CCV14] Silvia Cateni, Valentina Colla, and Marco Vannucci. A method for resampling imbalanced datasets in binary classification tasks for real-world problems. *Neurocomputing*, 135 :32–41, 2014.

[CHA14] Magali CHAMPION. *Contribution à la modélisation et l'inférence de réseau de régulation de gènes*. PhD thesis, Université Toulouse 3 Paul Sabatier, 2014.

[CJK04] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial : special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1) :1–6, 2004.

[CJM06] Mark Culp, Kjell Johnson, and George Michailides. ada : An r package for stochastic boosting. *Journal of Statistical Software*, 17(2) :1–27, 9 2006.

[Die00] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[dMDL11] PierreLafaye de Micheaux, Rémy Drouilhet, and Benoît Liquet. Présentation du logiciel r. In *Le logiciel R*, Statistique et probabilités appliquées, pages 1–6. Springer Paris, 2011.

[Dom99] Pedro Domingos. Metacost : A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164. ACM, 1999.

[EdSMdSJ00] J. Estrela da Silva, J.P. Marques de Sá, and J. Jossinet. Classification of breast tissue by electrical impedance spectroscopy. *Medical and Biological Engineering and Computing*, 38(1) :26–30, 2000.

[ET94] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[FS99] Yoav Freund and Robert Schapire. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780) :1612, 1999.

[FSZC] Wei Fan, Salvatore J Stolfo, Junxin Zhang, and Philip K Chan. Adacost : misclassification cost-sensitive boosting.

[GEN10] Robin GENUER. *Forêt aléatoire : aspects théorique, selection de variales et application*. PhD thesis, Université paris sur11, 2010.

[GFT+12] Mikel Galar, Alberto Fernández, Edurne Barrenechea Tartas, Humberto Bustince Sola, and Francisco Herrera. A review on ensembles for the class imbalance problem : Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(4) :463–484, 2012.

[GHCH11] Ming Gao, Xia Hong, Sheng Chen, and Chris J Harris. A combined smote and pso based rbf classifier for two-class imbalanced problems. *Neurocomputing*, 74(17) :3456–3466, 2011.

[GSM12] Vicente García, Javier Salvador Sánchez, and Ramón Alberto Mollineda. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, 25(1) :13–21, 2012.

[HAF13] GAHDOUM HAFIDA. Classification des donnée déséquilibréemédicale. Master's thesis, Université de tlemcen, 2013.

[Hao14] Wang Yanli Bryant Stephen H. Hao, Ming. An efficient algorithm coupled with synthetic minority over-sampling technique to classify imbalanced pubchem bioassay data. *Analytica Chimica Acta*, 806(Complete) :117–127, 2014.

[HB97] Thien M. Ha and Horst Bunke. Off-line handwritten numeral string recognition. *Pattern Recognition*, 31 :257–272, 1997.

[HC10] T Ryan Hoens and Nitesh V Chawla. Generating diverse ensembles to counter the problem of class imbalance. In *Advances in Knowledge Discovery and Data Mining*, pages 488–499. Springer, 2010.

[HWB14] Ming Hao, Yanli Wang, and Stephen H Bryant. An efficient algorithm coupled with synthetic minority over-sampling technique to classify imbalanced pubchem bioassay data. *Analytica chimica acta*, 806 :117–127, 2014.

[IG96] Ross Ihaka and Robert Gentleman. R : a language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3) :299–314, 1996.

[JKA01] Mahesh V Joshi, Vipin Kumar, and Ramesh C Agarwal. Evaluating boosting algorithms to classify rare classes : Comparison and improvements. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 257–264. IEEE, 2001.

[JN84] Nuggehally S Jayant and Peter Noll. Digital coding of waveforms : principles and applications to speech and video. *Englewood Cliffs, NJ*, pages 115–251, 1984.

[Lic13] M. Lichman. UCI machine learning repository, 2013.

[Liu04] Alexander Yun-chung Liu. *The effect of oversampling and undersampling on classifying imbalanced text datasets*. PhD thesis, Citeseer, 2004.

[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2) :212–261, 1994.

[LYCL10] CY Lee, MR Yang, LY Chang, and ZJ Lee. A hybrid algorithm applied to classify unbalanced data. In *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*, pages 618–621. IEEE, 2010.

[Mar08] Simon Marcellin. *Arbres de décision en situation d'asymétrie*. PhD thesis, Université Lumière Lyon II, le 2 Septembre 2008.

[RD13] M Mostafizur Rahman and DN Davis. Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2) :224–228, 2013.

[RS13] Oliver Rösler and David Suendermann. A first step towards eye state prediction using eeg. *Proc. of the AIHLS*, 2013.

[SKHN10] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost : A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 40(1) :185–197, 2010.

[Suc06] Henri-Maxime Suchier. *Nouvelles Contributions du Boosting en Apprentissage Automatique*. PhD thesis, université Jean Monnet de Saint-Etienne, 2006.

[Tor10] L. Torgo. *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010.

[VC06] Florian Verhein and Sanjay Chawla. Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In *Database Systems for Advanced Applications, 11th International Conference, DASFAA 2006, Singapore, April 12-15, 2006, Proceedings*, pages 187–201, 2006.

[Wei04] Gary M. Weiss. Mining with rarity : A unifying framework. *SIGKDD Explor. Newsl.*, 6(1) :7–19, June 2004.

[WY09] Shuo Wang and Xin Yao. Diversity analysis on imbalanced data sets by using ensemble models. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 324–331. IEEE, 2009.

[XLNY09] Yaya Xie, Xiu Li, EWT Ngai, and Weiyun Ying. Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36(3) :5445–5449, 2009.

[YL09] Show-Jane Yen and Yue-Shi Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3) :5718–5727, 2009.

[YMD+13] Zhang Yongqing, Zhu Min, Zhang Danling, Mi Gang, and Ma Daichuan. Improved smotebagging and its application in imbalanced data classification. In *Conference Anthology, IEEE*, pages 1–5, Jan 2013.

[YZZJ14] Qing-Yan Yin, Jiang-She Zhang, Chun-Xia Zhang, and Nan-Nan Ji. A novel selective ensemble algorithm for imbalanced data classification based on exploratory undersampling. *Mathematical Problems in Engineering*, 2014, 2014.

[ZIR13] Brice ZIRAKIZA. ForÊts alÉatoires pac-bayÉsiennes. Master's thesis, Université laval, 2013.

Résumé

Une base de données déséquilibrée est une base dont l'effectif d'une classe est largement plus petit en comparaison avec les autres classes présentes.

Dans ce projet de fin d'études de Master, nous proposons le traitement des données déséquilibré par les méthodes d'ensemble. De ce fait, notre contribution va être dans les trois points suivant : en premier lieu, nous appliquons 2 types de méthodes d'ensemble (Bagging, Adaboost) et leur hybridation avec les méthodes d'échantillonnage (SMOTEBagging, SMOTEBoost, RUSBoost, UnderBagging) sur différentes bases de données médicales. En second lieu, nous étudions l'influence de l'approche SMOTE avec différents degrés de déséquilibres sur le classifieur adaboost. En dernier lieu, nous évaluons la combinaison deux types d'échantillonnage (SMOTE, RUS) avec adaboost.

Les résultats empiriques montrent que l'hybridation de SMOTE avec la méthode d'ensemble Adaboost est plus performante que les autres méthodes testées.

Mots clés : données déséquilibrées, méthodes d'ensemble, méthodes d'échantillonnage, SMOTE, Adaboost.

Abstract

An unbalanced database is a base whose class size is largely smaller in comparison to other classes present.

In this master's project, we propose data processing unbalanced by ensemble methods. Therefore, our contribution will be within three points : first, we apply two types of ensemble methods (Bagging, Adaboost) and hybridization with Sampling (SMOTEBagging, SMOTEBoost, RUSBoost, UnderBagging) on different medical databases. Secondly, we study the influence of Smote approach with different degrees of imbalances AdaBoost classifier. Finally, we evaluate the combination of two types of sampling (smote, RUS) with AdaBoost.

The empirical results show that hybridization smote with the ensemble method Adaboost is more efficient than other tested methods.

Keywords : umbalanced data, ensemble methods, samling methods, SMOTE, Adaboost.